COMPUTING
edge

EMERGING TECHNOLOGIES

Also in this issue:

> Changing Trust

> The Power to Create Chaos

OCTOBER 2016

www.computer.org

IEEE

IEEE computer society

CELEBRATING 70 YEARS

2017

Richard E. Merwin Award for Distinguished Service

CALL FOR AWARD NOMINATIONS

Deadline 15 October 2016

ABOUT THE MERWIN AWARD

The highest level volunteer service award of the IEEE Computer Society for outstanding service to the profession at large, including significant service to the IEEE Computer Society or its predecessor organizations.

ABOUT RICHARD MERWIN

Richard Merwin was a pioneer in digital computer engineering who participated in the development of the ENIAC, MANIAC, and STRETCH computers. Despite a busy and productive technical career, Merwin found time to be active in professional societies, including the IEEE Computer Society, ACM and AFIPS. His generosity of spirit and genuine helpfulness was an important element in the progress of the computer profession.

AWARD

A bronze medal and $5,000 honorarium are awarded.

PRESENTATION

The Richard E. Merwin Award is presented at the IEEE Computer Society's Annual Awards Ceremony.

REQUIREMENTS

This award requires 3 endorsements.

NOMINATION SUBMISSION

Nominations are being accepted electronically www.computer.org/web/awards/merwin

MICHAEL R. WILLIAMS

2016 Richard E. Merwin Award for Distinguished Service

In recognition of exemplary contributions to the computing profession through outstanding scholarship and enduring and committed volunteer participation and leadership of the IEEE Computer Society.

AWARDS HOMEPAGE
www.computer.org/awards

CONTACT US
awards@computer.org

COMPUTING edge

IEEE COMPUTER SOCIETY http://computer.org • +1 714 821 8380

Printed with inks containing soy and/or vegetable oils

IEEE

IEEE computer society
CELEBRATING 70 YEARS

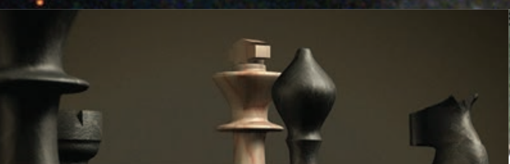
BPA WORLDWIDE

Circulation: ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

OCTOBER 2016 • VOLUME 2, NUMBER 10

COMPUTING edge



Departments

29

Displays as a Material: A Route to Making Displays More Pervasive

Modern Computer Arithmetic

Paolo Montuschi, Polytechnic University of Turin

Jean-Michel Muller, Centre National de la Recherche Scientifique

This installment highlighting the work published in IEEE Computer Society journals comes from IEEE Transactions on Computers.

A 2009 *IEEE Transactions on Computers* (*TC*) guest editorial called computer arithmetic "the mother of all computer research and application topics." Today, one might question what computer arithmetic still offers in terms of advancing scientific research; after all, multiplication and addition haven't changed. The answer is surprisingly easy: new architectures, processors, problems, application domains, and so forth all require computations and are open to new challenges for computer arithmetic. Big data crunching, exascale computing, low-power constraints, and decimal precision are just a few domains in which advances are implicitly pushing for rapid, deep reshaping of the traditional computer-arithmetic framework. *TC* (www.computer.org/web/tc) has long published regular submissions as well as special sections on this topic, including one scheduled for 2017. Here, we focus on three recently published papers.

In "Parallel Reproducible Summation," James Demmel and Hong Diep Nguyen (*IEEE Trans. Computers*, vol. 64, no. 7, 2015, pp. 2060–2070) address result reproducibility in cases where it's a requirement. They present a technique for floating-point reproducible addition that doesn't depend on the order in which operations are performed, which makes it appropriate for massively parallel environments.

Mioara Joldeş and her colleagues deal with manipulation of floating-point expansions in "Arithmetic Algorithms for Extended Precision Using Floating-Point Expansions" (*IEEE Trans. Computers*, vol. 65, no. 4, 2016, pp. 1197–1210). Such expansions, which are unevaluated sums of a few floating-point numbers, might be used when one temporarily needs to represent numerical values with a higher precision than that offered by the available floating-point format. The authors introduce and prove new algorithms for dividing and square-rooting floating-point expansions, as well as for "normalizing" such expansions.

In "On the Design of Approximate Restoring Dividers for Error-Tolerant Applications" (*IEEE Trans. Computers*, vol. 65, no. 8, 2016, pp. 2522–2533), Linbin Chen and his colleagues propose several approximate restoring-divider designs. Their simulation results show that, compared with nonrestoring division schemes, their designs had superior delay, power dissipation, circuit complexity, and error tolerance. Most striking, the approximate designs offer better error tolerance "for quotient-oriented applications (image processing) than remainder-oriented applications (modulo operations)."

These papers are a small but representative view of trends in computer arithmetic. However, computer arithmetic also bridges the gap between architecture and application design—and thus will continue to advance in vibrant directions, provided it maintains strong connections with technology and advanced research.

PAOLO MONTUSCHI is a professor of computer engineering at the Polytechnic University of Turin. Contact him at pmo@computer.org or visit http://staff.polito.it/paolo.montuschi/news-from-EIC-TC.html.

JEAN-MICHEL MULLER is a Senior Researcher at Centre National de la Recherche Scientifique (National Center for Scientific Research), École Normale Supérieure, Lyon. Contact him at jean-michel.muller@ens-lyon.fr or visit http://perso.ens-lyon.fr/jean-michel.muller.

 2469-7087/16/$33.00 © 2016 IEEE

CS FOCUS

Magazine Roundup

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to cloud migration and microchip manufacturing. Here are highlights from recent issues.

Computer

As systems based on silicon and conventional architecture reach their limits, researchers are exploring and harnessing **new computing paradigms**, a subject explored in *Computer*'s September 2016 special issue.

IEEE Software

IEEE Software's September/October 2016 issue has two theme topics. Articles on the **business of software** address subjects such as better selection of software providers via trial-sourcing, business roles in the emerging open data ecosystem, and offshoring's hourly cost. Articles on the **software engineering process** cover subjects including agile software development's risks, moving from software development to software assembly, choosing software-development methodologies, and steering software-development workflow.

IEEE Internet Computing

According to Internet pioneer Vint Cerf in "On Risk," from *IEEE Internet Computing*'s July/August 2016 issue, as we move toward the Internet of Things era, loss of security and privacy becomes a greater risk. This will make **responsible programming** more of a concern. Cerf says it's only a matter of time until programmers are held accountable for their work's quality and safety, and he asks how the software industry will plan for this.

Computing in Science & Engineering

Cloud-computing advances are increasingly making data analysis available as a service. This approach accelerates the discovery process by letting researchers focus on scientific discovery, while computational experts provide the enabling services. The **science-as-a-service** process also reduces research costs, permits rapid hypothesis testing and exploration, makes scientific software sustainable, and lowers entry barriers to large-scale scientific analysis. *CiSE*'s September/October 2016 special issue explores these matters.

IEEE Security & Privacy

Software-defined mobile network (SDMN) architecture integrates software-defined networks, network virtualization, and cloud computing in mobile environments to make legacy networks scalable and dynamic. However, because the architecture separates the control and data planes, it significantly changes mobile-security management and application. The authors discuss this in “Opportunities and Challenges of Software-Defined Mobile Networks in Network Security,” from *IEEE S&P*’s July/August 2016 issue.

IEEE Cloud Computing

Organizations are increasingly collecting data via Internet of Things sensing devices and shipping it to remote cloud datacenters for analysis. Ensuring end-to-end security as the data travels between edge and cloud datacenters is challenging. This is the focus of “**Threats to Networking Cloud and Edge Datacenters** in the Internet of Things,” which appears in *IEEE Cloud Computing*’s May/June 2016 issue.

IEEE Computer Graphics and Applications

Visual text analysis as a way to convert textual data into actionable knowledge has become a popular research subject. In “Topic- and Time-Oriented Visual Text Analysis,” from *CG&A*’s July/August 2016 issue, the authors present the benefits and challenges of combining text analysis with interactive visualization.

I*EEE Intelligent Systems*

A trend in human–computer interaction is using intelligent systems to make creating apps, games, and other content easier, letting even nonskilled users define a system’s behavior. “**Semantics-Based Intelligent Human-Computer Interaction**,” from *IEEE Intelligent Systems*’ July/August 2016 issue, addresses this approach’s challenges by proposing a framework for automatically configuring a system’s behavior based on user input and contextual information.

IEEE MultiMedia

People’s abilities change over time due to many factors, which affects their usage of multimedia computing. In addition, individuals with disabilities have specific requirements that necessitate a person-centered, adaptive approach. In “**Person-Centered Multimedia Computing**: A New Paradigm Inspired by Assistive and Rehabilitative Applications,” from *IEEE MultiMedia*’s July–September 2016 issue, the authors explain their solution to the problem.

IEEE Annals of the History of Computing

IEEE Annals’ July–September 2016 theme issue on **network origins and evaluations** includes articles on topics such as the restructuring of Internet standards governance between 1987 and 1992; infrastructure, representation, and historiography in BBN’s Arpanet Maps; and two early interactive-network experiments.

I*EEE Pervasive Computing*

Lightweight, highly autonomous drones that interact with the world are emerging and will dramatically change consumer electronic technology, in the same way that smartphones revolutionized personal computing. The authors of “Flying Smartphones: When Portable Computing Sprouts Wings,” from *IEEE Pervasive Computing*’s July–September 2016 issue, explore this subject.

IT Professional

“Addressing Pressing Cybersecurity Issues through Collaboration,” which appears in *IT Pro*’s July/August 2016 issue, discusses this year’s projects of the US National Institute of Standards and Technology’s **National Cybersecurity Center of Excellence**. The center works with the private sector, academia, and other government organizations to identify pressing cybersecurity problems.

IEEE Micro

IEEE Micro’s July/August 2016 special issue presents four articles chosen from the 2015 **IEEE Symposium on High Performance Interconnects**.

Computing Now

The Computing Now website (http://computingnow.computer.org) features **up-to-the-minute computing news** and blogs, along with articles ranging from peer-reviewed research to opinion pieces by industry leaders.

EDITOR'S NOTE

Tracking Emerging Technologies

The world of technology is always changing, as existing approaches are improved upon and new approaches are developed. To remain relevant in their fields, professionals must keep up with emerging technologies. This *ComputingEdge* issue focuses on some of these important developments.

Using Internet of Things technology in healthcare promises to significantly improve patients' well-being while alleviating the problem of scarce resources. "The Internet of Things in Healthcare: Potential Applications and Challenges," from *IT Professional*, reviews some of the most promising applications and the significant challenges their implementation faces.

Many participants in November's US presidential election will be using electronic voting machines. The author of *Computer*'s "Coda in the Key of F2654hD4" says it's time to re-examine these machines' security.

Developers of systems of systems (SoSs) face numerous challenges. In response, the authors of "Monitoring Requirements in Systems of Systems," from *IEEE Software*, are developing the ReMinds tool, which instruments the systems in an SoS to extract events and data at runtime. It then defines requirements as constraints to check for expected behavior and properties.

Safely operating drones presents communication and computational challenges. *IEEE Internet Computing*'s "An Internet of Drones" looks at these challenges and describes a prototype system that addresses them via novel network and software architectures.

"Ten Open Questions for Techno-Optimists," from *IEEE Micro*, discusses some of the questions regarding productivity growth and economic gains resulting from innovative IT.

If privacy science is in enough of a crisis that resolution requires a paradigm change, then a scientific revolution is upon us, according to "Privacy's Paradigm," from *IEEE Security & Privacy*.

The authors of *IEEE Pervasive Computing*'s "Displays as a Material: A Route to Making Displays More Pervasive" advocate changing to an architecture that relies on autonomous pixels that independently sense input and convert it to a visual output. They also discuss their two display prototypes.

ComputingEdge articles on topics other than emerging technologies include the following:

- Public-key cryptography turns the nature of online trust into a mechanical reaction, explains the author of *Computer*'s "Changing Trust."
- According to the author of "The Power to Create Chaos," from *Computing in Science & Engineering*'s July/August 2016 issue, computers are the only research tools that, by design, exhibit chaotic behavior in which a minimal change to a computation's input can significantly change its output. He says scientific-software developers and users should be aware of this and set up safety nets for protecting themselves against unfortunate surprises.


C 100
Focus on
Your Job Search
IEEE Computer Society Jobs helps you easily find a new job in IT, software development, computer engineering, research, programming, architecture, cloud computing, consulting, databases, and many other computer-related areas.
New feature: Find jobs recommending or requiring the IEEE CS CSDA or CSDP certifications!
Visit www.computer.org/jobs to search technical job openings, plus internships, from employers worldwide.
http://www.computer.org/jobs
IEEE computer society | JOBS
The IEEE Computer Society is a partner in the AIP Career Network, a collection of online job sites for scientists, engineers, and computing professionals. Other partners include Physics Today, the American Association of Physicists in Medicine (AAPM), American Association of Physics Teachers (AAPT), American Physical Society (APS), AVS Science and Technology, and the Society of Physics Students (SPS) and Sigma Pi Sigma.

Privacy's Paradigm

Daniel E. Geer Jr.
In-Q-Tel

In his 1962 landmark book, *The Structure of Scientific Revolutions*, T.S. Kuhn begins and ends with the circular idea that a scientific community is defined by the beliefs practitioners share, and those beliefs define what community the practitioners belong to. This is, in fact, instructive, as part of becoming a mature science is the development of a broad consensus about the core concerns of that branch of knowledge. Kuhn's word for the collections of exemplars of good science was *paradigm*, a word whose meaning today is all but entirely Kuhn's, even among those who've yet to read his masterwork.

But what is a paradigm, and why do we want one? As Kuhn put it, "[Paradigms] are the source of the methods, the problem field, and the standards of solution accepted by any mature scientific community at any given time." Simply put, a paradigm is all the things a scientist can assume that his or her colleagues will congenially understand about their common work without having to explain them or argue them from first principles again and again.

As everyone knows, from time to time a science might undergo a revolution, which in Kuhn's terms is precisely the laying down of one paradigm in preference for another. The title of his book is to be understood as precisely that: scientific revolutions share aspects of structure that we can now describe, because there have been enough of them in the past 400 years to discern that structure. If you consider physics to be the paragon of a hard science, then the transition from Newtonian mechanics to Einsteinian relativity demonstrates exactly the point Kuhn was making—that there comes a moment when research has reached a kind of impasse where the nature of what seem to be puzzles needing further study can't be profitably investigated within the paradigm that now holds.

Kuhn referred to these impasses as the appearance of an anomaly, a puzzle that we can't evaluate by way of further research consistent with the existing paradigm. His review of past revolutions centered in each case on the appearance of irreconcilable anomalies that made a given field ripe for revolution. (For example, that roasting metals caused them to gain weight, indicating that they had absorbed some fraction of the air around them—a fraction that could be exhausted—led to the idea that air might not be the one and only gas but rather a combination of gases.)

When there's a paradigm shift—that is to say a scientific revolution—it may serve to redirect a field so completely that some parts of it fall away entirely. The separation of astronomy from astrology, or of chemistry from physics, are two examples. As Kuhn put it, the choice between paradigms is a choice between incompatible modes of community life. And, yes, he does mean "scientific community" in an altogether social sense: "Since no two paradigms leave all the same problems unsolved, paradigm debates involve the question: which problems is it more significant to have solved?"

Cybersecurity has long had the paradigm of perimeter control, but that paradigm is in total crisis.[1] The paradigm that's the obvious alternative to perimeter control (and thus authentication as its gating function) is accountability based on one single unspoofable identity per person. Real soon now, identity will simply be an observable that needs no assertions—because everything we are or do is unique if examined closely enough, then verifying an assertion like "my name is Dan" can easily morph into an observable like "sensors say that this is Dan." The National Strategy for Trusted Identities in Cyberspace—the 2011 US government initiative to improve the privacy and security of online transactions—isn't worded in that way, though that's how I read it. That means a crisis in privacy science, too.

Privacy's paradigm has long been selective revelation,[2] or as I prefer, a state of privacy is the effective capacity to misrepresent oneself. But, as is obvious, being part of the modern world in no more robust way than appearing unmasked on a public street is the same as submitting to unitary identification

2469-7087/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

This article originally appeared in IEEE Security & Privacy, *vol. 14, no. 4, 2016.*

observable at a distance by things you've never heard of. Ergo, the individual's choice is between submission or withdrawal—it doesn't include selective revelation.

The crisis around the paradigm of selective revelation is that (as with metadata) there's so much redundancy in what's observable that prohibiting one or another form of collection has no meaningful effect whatsoever on those agencies, whether intelligence or advertising, who would build a model of you from observability alone. As an example, with current technology I can read the unique radio signature of your beating heart at five meters. As with anything that has an electromagnetic output, the only technologic question is the antenna quality. If I can take your picture on a public street without your permission or notice, why can't I record your heart? Or your iris? Or your gait? Or the difference in temperature between your face and your hands? That list is long and getting longer. It's a crisis for which the paradigm of selective revelation can scarce put up puzzles fast enough, and scientific solving of those puzzles trails the curve.

The crisis is simply that what heretofore we have known as confidentiality is becoming quaint and irrelevant, and not merely because all uses of virtual reality come with total surveillance. Perhaps privacy science will have to reposition confidentiality within some new paradigm that prioritizes integrity, not confidentiality. Perhaps a world in which data can and will be collected irrespective of selective permission granting is a world in which the data had better be right. If more and more intelligent actors are to be out there doing our implicit bidding, then data integrity had better be as absolute as we can make it, and that's where the research puzzles will have to be found.

Kuhn took some pains to say why it is that a paradigm shift requires a crisis, that "to an extent unparalleled in other fields, [scientists] have undergone similar educations and professional initiations." One must ask the central question of this essay by mirroring Kuhn: Is the paradigm of privacy science in enough of a crisis that resolution of the crisis requires a change of paradigm? If the crisis is sufficient to require a reformulation of the paradigm of privacy, then a scientific revolution is upon us—what Kuhn calls "a reconstruction of group commitments." As he points out, a crisis requiring such a reconstruction might not even originate within privacy research itself, but instead be due to discoveries in some other field or venue, just as discoveries in physics engendered a crisis in chemistry once upon a time.

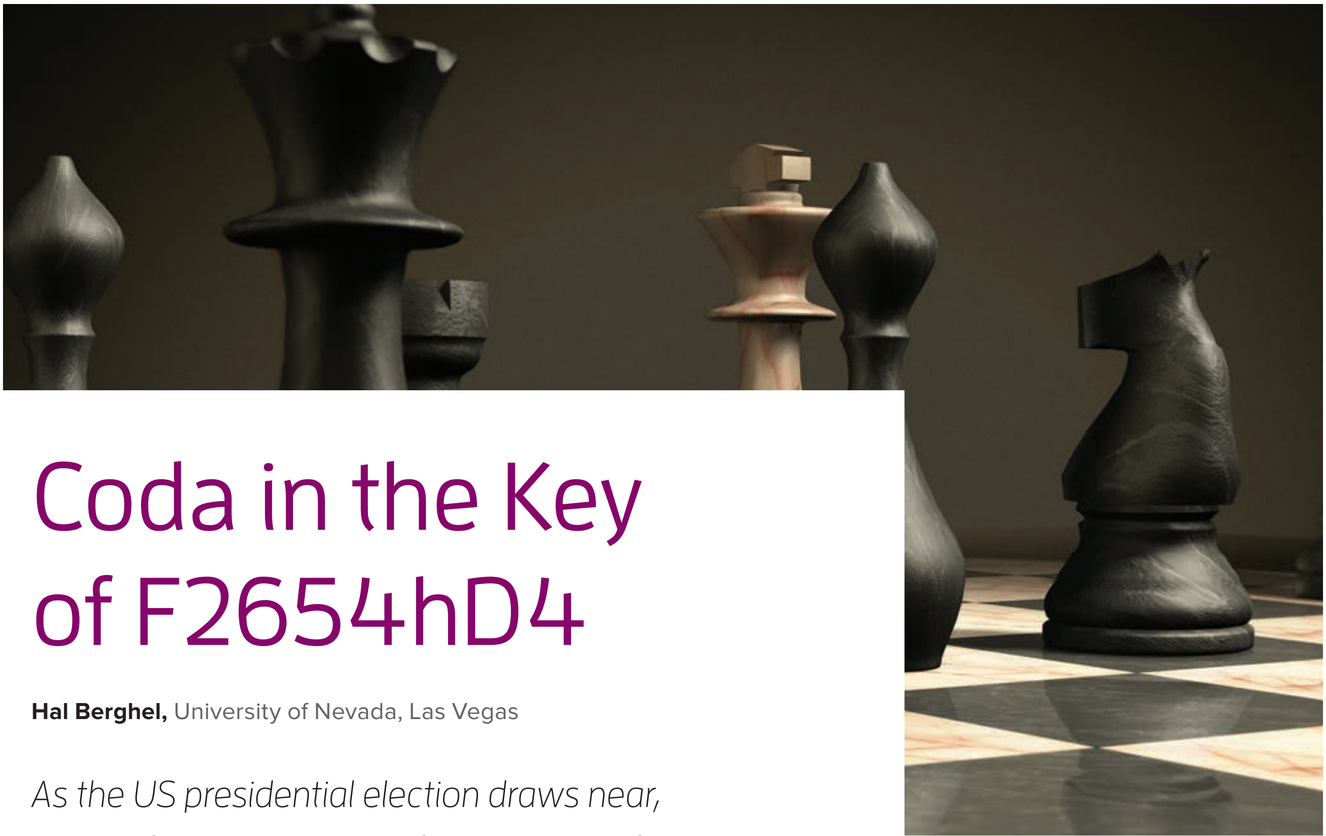
So it is now.

A fuller treatment of this topic can be found at geer.tinho.net/geer.nsf.6i15.txt. ■

References

1. D. Geer, "The Shrinking Security Perimeter," audio mirror, 2004; geer.tinho.net/Dan_Geer_-_The_Shrinking_Security_Perimeter.mp3.
2. E. Hughes, "Privacy is the Power to Selectively Reveal Oneself to the World," *A Cypherpunk's Manifesto*, 1993; www.activism.net/cypherpunk/manifesto.html.

Daniel E. Geer Jr. is the chief information security officer of In-Q-Tel. Contact him at dan@geer.org.



Coda in the Key of F2654hD4

Hal Berghel, University of Nevada, Las Vegas

As the US presidential election draws near, many of us can expect to find ourselves face to face with an electronic voting machine. It's time to re-examine the integrity of these machines.

Remember that default DES key that was hard coded in the Diebold voting machines for many years? Despite revelations that shocked the voting public, computer scientists concerned with the systems security of Direct-Recording Electronic (DRE) voting machines were central to uncovering the now familiar Diebold debacle of 2005—when activist Bev Harris discovered and then posted the unprotected source code for the Diebold TS and TSx voting machines. She recovered the code from the Diebold website.[1] What unfolded was a fascinating chronicle of corporate irresponsibility, hubris, incompetence, political chicanery, and power politics—all wrapped up in a story befitting a good dime store novel. And the proverbial plot thickened when computer scientists got involved—at that point, things got downright ugly. As painful as it was for the computer scientists involved, the country is far better off for it.

STANDARDS VACUUMS

The professional computing community is very familiar with the role of standards. Well-known standards, such as ISO 17799 for IT security and the ISO 9000 series for management, establish guidelines and general principles that codify industry best practices. In some cases independent certification bodies are used to assure customers and the public of compliance. Although we all work with standards differently, we can agree on two things: first, standards provide a minimal assurance of integrity and quality, and second, wandering too far afield usually comes at a cost in terms of safety, reliability, performance, profitability, and credibility for the affected organization and its representatives.

There are standards for quality, safety, reliability, and so on, in industries related to food, drugs, military equipment, manufacturing, computer equipment, software, household appliances, floor coverings, and paint, just to name a few. However, one area that's historically been immune to reasonable standards is the manufacturing and use of voting equipment—that which determines our political future.

Published by the IEEE Computer Society

EDITOR HAL BERGHEL
University of Nevada, Las Vegas; hlb@computer.org

The voting franchise has been operating in a standards vacuum for decades, and this vacuum extends well beyond voting equipment.[2] The Diebold story is just the tip of the iceberg.

We know that using the same encryption key for every transaction hasn't been an acceptable practice since the Caesar Cipher was popular in ancient Rome. Furthermore, since the mid-1970s, the DES algorithm was known to be vulnerable to brute-force attacks because of its short key length.[3,4] After DES was deprecated by the National Institute of Standards and Technology (NIST) and replaced by the Advanced Encryption Standard (AES), Diebold went on to hard code it into the source code—a willful circumvention of best practices for the sake of cost savings and expedience. That they then lied about it was an even greater betrayal of the public trust. All of this was possible because of the lack of both industry standards and accountability to the public. Further, Diebold carried on this way for an entire generation of voting machines.

Diebold's story is a shining example of the voting machine industry's heritage of stupidity and arrogance and the public's tolerance of proprietary electronics and software that have never been adequately tested by impartial, legitimate domain experts. Harris's disclosure broke the proprietary veil, and some daring computer scientists read the curious public in on some outrageous security breaches that we never would have known about.

Exposure of the Diebold AccuVote system's weakness is generally credited to Johns Hopkins University computer scientist Aviel Rubin and his colleagues, who in 2003 began analyzing the source code discovered by Harris.[1] It's useful to frame this story in terms of Rubin's analysis of the Diebold source code. Here's what he found:

1. The AccuVote system anonymized the voting order with a linear congruential generator (LCG) that didn't work properly and was inappropriate for this purpose, thereby undercutting the principle of the secret ballot.
2. In parts of the code that required cryptography, either the algorithms were incorrectly applied or not used at all. (Of course, as mentioned, DES was used despite having been deprecated by NIST.)
3. Diebold's approach to key management was juvenile. The same encryption key (see column title) was hardcoded into every voting machine. The vulnerability stemming from a lack of key management was first reported in 1997 by University of Iowa computer scientist Douglas W. Jones without effect (http://homepage.cs.uiowa.edu/~jones/voting/dieboldacm.html).
4. The association between a candidate's record in the ballot definition file and the appearance on the screen of the AccuVote DRE voting machine wasn't cryptographically protected, thus verifying that a voter's intent was accurately recorded was impossible. The fact that a vote appeared in the confirmation screen (front end) was no guarantee that that vote was recorded and tabulated (back end). Diebold's "redundant" storage technique only ensured duplicate copies of the voting transaction, not that any of the transactions were correct.
5. The ballot definition file contained sensitive information like the terminal's ID number, the dial-in numbers for online tally reports, IP addresses for networked computers, and user names and passwords—all in plaintext.
6. The smartcards used by voters to authenticate with the voting machines used no cryptography at all. Therefore, anyone with the ability to create smart cards offsite and get them inserted into an AccuVote DRE station (trivial—see below) could authenticate to the machine and have votes recorded.
7. Election officials' administrative cards all had a default PIN of 1111.

Note that these are professional, technical comments—not parochial or political opinions. Most of us recognize the faults as rookie mistakes that wouldn't withstand scrutiny in a respectable college-level computer science programming class. Rubin claims in his book that this level of sloppiness was characteristic of the entire Accuvote TS code base. For making these deficiencies known, he was vilified by Diebold, sundry election officials, and an occasional politician. Such blowback against technical experts was repeated several times before the Diebold story had played out. We'll pass over the idiocy of making unprotected source code available

The Diebold debacle is fascinating chronicle of corporate irresponsibility, hubris, incompetence, political chicanery, and power politics.

through Diebold's website in silence. The point to remember is that Rubin's was the same low assessment of source code that any of us would give to our students. It was an accurate, fair, and legitimate criticism of sloppy work.

However, Rubin wasn't the Diebold code's only critic. A series of investigations by computer security specialist, Harri Hursti, proved to be even more embarrassing.

THE HURSTI HACKS

In the mid-2000s, Hursti conducted several evaluations of the Diebold AccuVote system on behalf of Black Box Voting, Harris's election activist organization. The analysis was fairly extensive, pointing to deficiencies in the Diebold boot loader, removable memory, and easy-to-circumvent hardware security. Hursti correctly assessed Diebold's three-layer architecture as insecurity-in-depth, which is one step below security through obscurity.[5] Without question, the most alarming insecurity had to do with Diebold's removable memory cards. In the literature, these are the most prominent of the so-called Hursti hacks. These hacks were so simple and dramatic they were featured in the 2006 HBO documentary, *Hacking Democracy* (http://hackingdemocracy.com).

Hursti showed that the insecurities of the AccuVote TS and AccuVote TSx OSs were so substantive that even elementary changes to the code and/or data fields on the removable memory cards could change the outcome of elections. Despite the fact that these hacks had been demonstrated several times in several different jurisdictions, the initial response from Diebold was to attack the messenger(s). They even demanded that HBO cancel the previously mentioned documentary, but without effect.

Remember that Rubin published his analysis of the source code and Hursti followed up with experimental demonstrations of some result-altering hacks. Diebold's defense and counterclaims began to permanently unravel when University of California and then Princeton University researchers confirmed Hursti's results.[6,7] An earlier independent review by the Science Applications International Corporation (SAIC)—commissioned by the State of Maryland in 2003—had also reported that the "AccuVote-TS voting system is not compliant with the State of Maryland Information Security Policy and Standards … and is at a high risk of compromise."[8]

A follow-up review by Maryland-based RABA Technologies, LLC, echoed the SAIC report, and added that the back-end management system (GEMS) was also insecure. So by the time *Hacking Democracy* came out, Diebold's proverbial cat was separated from the bag by light years. Diebold's response to these revelations was typical of the power elite: "… voters in the state of Maryland can now rest assured that they will participate in highly secure and accurate elections." Then-governor Robert L. Ehrlich Jr. (R), opined that "Because of this [SAIC] report, Maryland voters will have one of the safest election environments in the nation" (both quotes appear in *Brave New Ballot*,[1] pp. 137–138). There's no way to know whether the better explanation of Diebold's and Ehrlich's spin is cognitive dissonance or outright deceit, but whatever the reason, the known code insecurities went unattended for many years.

Although the hacks themselves are of only marginal historical significance at this point, the complex interplay among Diebold, the election officials who either tried to cover up the insecurities or expose them, the politicians who sought political cover from the exposures, and the computer scientists who uncovered the problems remains critical for little has changed to correct the problems. The capacity of the manufacturers, vendors, and election officials to conceal, cover up, and deceive is as great today as it was 15 years ago. But more is at stake now because DRE voting machines are ubiquitous, and we've since developed a tolerance for chicanery in our elections. There's another player that I haven't mentioned: the Independent Testing Authorities (ITAs) that "validate" these voting systems.

ITAs AND VOTER "VERIFICATION"

Diebold, Sequoia, and Election Systems and Software (ES&S) came to dominate the digital voting equipment market by the early 2000s. After a few mergers and acquisitions cycles, Diebold and Sequoia became subsidiaries of Dominion Voting Systems. At this point, the competition has been narrowed to a very few players.

Once a voting system is developed, election officials might be deluded into a false sense of security by ITAs (now called Voting System Testing Laboratories) that certify the system's integrity. ITAs work in much the same way as credit ratings services (think Moody's, Standard and Poor's, and the Fitch Group), and they're bound by the same incentives. In all cases, the applicant pays for the service of certification—that is, the beneficiary of the certification provides the revenue stream to the certifier. Thus, if an ITA rejects certification of voting equipment (not likely), other ITAs are enlisted until one approves certification. This arrangement takes conflict of interest to a new plateau.

Further, ITAs/VSTLs only do extensional validation, which is to say they compare results by re-running election records with known outcomes or using canned datasets with well-defined data. That doesn't really contribute much confidence in the system if no one looks "under the hood." Chip design and circuit analysis aren't part of the validation because both are proprietary. No objective source code review is undertaken by skilled computer scientists, unless there's been an accidental leak like the one mentioned earlier. This incestuous relationship among ITAs, manufacturers, and technically ignorant election officials seeking to avoid public scrutiny of their activities is still with us today. This is why the Diebold story is still relevant. Although Diebold Election Systems and its amateurish source code is gone, the structural problems that gave rise to them in the first place are still with us.

If an ITA is to be effective, it must not only provide intentional validation that compares output from canned datasets, but it must also provide a functional analysis of the source code by impartial, skilled computer professionals. This is in effect what Rubin's, David Wagner's, and Ariel Feldman's teams did.[6,7] But at this writing, verification of voting systems' code might amount to nothing more than comparing hash signatures between file versions. Hash signatures are measures of binary identity—not the quality or integrity of code. (Primitive analysis restricted to I/O based on canned datasets is frequently referred to as black-box analysis, which is the source of the name Harris chose for her elections activist and investigatory group Black Box Voting.)

Further, Diebold apparently didn't bother to run their source code through a commercial-quality source code analyzer. However, Wagner and his colleagues did.[6] In the research they prepared for the California Secretary of State, the Fortify (now HP) static code analyzer identified 16 security vulnerabilities in the AccuVote operating system ranging from array bounds violations and faulty input validation errors to buffer overruns, buffer underruns, and pointer errors. Although the specific details (location of code fragment, and so on) were suppressed from the public report, enough detail was included to convince any computing professional that the code base lacked integrity. Note that these 16 vulnerabilities would have been easily detected with the HP Fortify static source code analyzer had Diebold chosen to use it. I encourage readers to review these referenced reports and confirm for themselves that the Diebold source code used up to and including the 2006 national elections should be on display in the Smithsonian as a primitive artifact. Any thoughtful analysis confirms the computer scientists' claims: the code was amateurish, the security standards were embarrassingly weak, and the systems were fraught with vulnerabilities. According to *Democracy Hacked*, approximately 40 percent of all votes in the US were counted by systems that ran this code at the time of the analysis.

> The Diebold source code used up to and including the 2006 national elections should be on display in the Smithsonian as a primitive artifact.

DON'T WORRY, BE HAPPY

Some have claimed that DRE voting machine security isn't an issue:[9]

> *The conjecture that ... we are unable to make such a simple system secure and accurate is contradicted by the facts of our everyday existence. We build secure and accurate computer systems that fly our airliners. We build secure and accurate computer systems that guide our submarines under the ice cap. We build secure and accurate computer systems that guide our astronauts to the moon and bring them safely back to earth. We submit to open heart surgery while a computer monitors our vital signs and controls an artificial heart and lung machine. The list of secure and accurate computer systems that monitor, control, and improve our lives is large and growing daily.*

The appropriate response to this argument is "that's true, but so what?" This is a patently silly position to take for a number of different reasons. For one, threat vectors must be understood in context. Who'd be incentivized to corrupt flight control systems? What might be gained if an airliner was off course? What relationship would the possible perpetrators likely have to the affected airlines? The same applies to moon landings, heart monitors, and the like. In each case, the likely threat would be terrorists or criminals, and external. The fear would be that someone outside the system wants to do harm to others and the incentive might be revenge, anger, hate, jealousy, greed, and so on—all motives that are visceral and personal.

Voting machines provide an entirely different context: the incentive is to subvert the democratic process toward partisan effect, and the likely perpetrator would be internal, or at least very closely linked to a specific political interest. Thus, the perp would likely be a partisan operative either employed by, or closely connected to, a candidate, party, PAC, or particular election officials. Murderers and terrorists tend to not work closely with domain knowledge experts on their weapons of choice. People that steal elections do.

Further, mission-critical systems rely on high-confidence software development paradigms. As shown, the Diebold code certainly wasn't high-confidence. If loss-of-life scenarios require high-confidence methodologies, loss-of-country scenarios make them similarly desirable. The Diebold DRE voting machines under discussion weren't trusted systems, rather, they're twisted systems in which minimal attention was paid to best practices in software development, software security, user privacy, software reliability, and so forth. The letter and spirit of industry standards in effect at the time the equipment was developed were violated, ensuring that expected results would be obtainable only under optimal circumstances in which all involved behaved properly and predictably and without serious corrupting external influences. Elections never offer such controlled environments.

In addition, as long as completely secret ballots are required, there's no way to fully and incontrovertibly audit the DRE voting machines' results. Even when paper-tape backup is used, the voting sequences are scrambled so the individual votes can't be recovered, and there's no voter verification of the vote, but only voter verification of the most recent behavior of the particular voting machine. Although some alternate voting systems have been proposed,[10] no commercial voting machines that I know allow each individual voter to verify that their vote was actually recorded by the digital vote management system and reported to state election officials correctly. For that level of assurance to result with current voting equipment, both the electronic and physical records have to be tallied and publicly reported independently. It's important to understand that the phrase "voter verified" normally refers to the voter's verification of the vote cast at the DRE voting terminal, not verification that it was recorded by the tally management system.

Murderers and terrorists tend to not work closely with domain knowledge experts on their weapons of choice. People that steal elections do.

WAG THE DOG

In most important respects, little was learned from the Diebold fiasco. To be sure, DRE voting machine manufacturers are more attentive to source code integrity, but the degree is a matter of conjecture because the code is still proprietary and not available for inspection—not even after the election concludes. Lack of trustworthy code remains a real and present danger to election integrity and our democracy. Vilification of the computer scientists wasn't due to their scientific results—no reasonable person challenged their facts—but because they cast doubt on the accuracy and honesty of the election results. In other words, they were castigated for pointing out the obvious: no insecure computing systems are trustworthy, DRE voting machines included!

In addition, the few electronic voting machine manufacturers that remain are more circumspect in how they represent their product to their customers—the jurisdictions and the public. Although Diebold's arrogance and hubris waned as the company headed toward collapse, there's still no mechanism through which the public can establish confidence in these manufacturers' products and practices—too much is hidden from view.

Thus, we can't know the degree to which obsolete, insecure code and bad practices are in active use. It's also unknown whether, or to what extent modern source code is built around a valid security model. In 2016, vendors and manufacturers can still manipulate election officials who lack technical knowledge and skills. Voting machine procurement and approval processes face partisan brinksmanship, and any election official demanding independent testing faces threat of litigation from manufacturers and perhaps even election officials. The overwhelming majority of jurisdictions still fail to perform complete audits of election results, and what's more, there's evidence to suggest that those who scrutinize the fairness of elections might be subject to government surveillance.[11] These aren't good signs.

At its inception, digital voting technology promised to promote universal suffrage—enabling underprivileged, disadvantaged, and immobile voters to come to the polls, as well as related benefits such as the minimization of pressure from partisans and some mitigation against the historical vote-suppression techniques (such as long wait times, confusing ballots, miscounts, undercounts, discarded ballots, corrupted results, and so on).[12] However, DRE voting machines haven't delivered on the promise to help secure the election franchise for all citizens. The Diebold scandal revealed that there's far too much slop in the digital voting process at too many different levels. For that understanding, we're indebted to the computer scientists mentioned in the studies I've referenced here. They're the true heroes of this story, and the country owes them a great debt.

All in all, the way electronic voting is administered in the US still falls far short of reasonable expectations in terms of the ability to verify election outcomes;[13] to technically validate the equipment's source code; to achieve a public understanding of the systems' vulnerabilities; and to completely disclose possible conflicts of interest between vendors and their agents, public officials, and those engaged in vetting the integrity and certification of voting systems. In these areas, we're no farther along than we were 20 years ago. Our best hope at addressing these problems rests with computing professionals. Indeed, our goal should be to demand that these experts be centrally involved in vetting all future voting systems. After all, our first line of defense is the community of computing professionals who are willing to take risks and speak out.

For those interested in this topic, I recommend the seminal book, *Broken Ballots: Will Your Vote Count?*[20] ◼

REFERENCES

1. A. Rubin, *Brave New Ballot*, Morgan Road Books, 2006.
2. H. Berghel, “Digital Politics 2016,” *Computer*, vol. 49, no. 1, 2016, pp. 75–79.
3. L. Hoffmann, “Q&A: Finding New Directions in Cryptography” [interview with Whitman Diffie and Martin Hellman], *Comm. ACM*, vol. 59, no. 6, 2016, pp. 110–112.
4. J. Gilmore, “DES (Data Encryption Standard) Review at Stanford University,” Toad Hall, 20 Sept. 2005 (with subsequent updates through 2015); www.toad.com/des-stanford-meeting.html.
5. H. Hursti, “Diebold TSx Evaluation—Critical Security Issues with Diebold TSx (unredacted),” security alert, Black Box Voting, 11 May 2006; www.blackboxvoting.org/BBVreportIIunredacted.pdf and Supplement www.blackboxvoting.org/BBVreportII-supplement-unredacted.pdf.
6. D. Wagner et al., “Security Analysis of the Diebold AccuBasic Interpreter,” Voting Systems Technology Assessment Advisory Board report, Presidential Commission on Election Administration, 14 Feb. 2006; https://web.archive.org/web/20070611092341.
7. A. Feldman, J.A. Halderman, and E. Felten, “Security Analysis of the Diebold AccuVote-TS Voting Machine,” *Proc. USENIX/ACCURATE Electronic Voting Technology Workshop* (EVT 07), 2007; www.usenix.org/legacy/events/evt07/tech/full_papers/feldman/feldman_html/index.html.
8. “Risk Assessment Report Diebold AccuVote-TS Voting System and Processes,” report commissioned by the Maryland Dept. of Budget and Management, Science Applications Int’l Corporation (SAIC), 2 Sept. 2003 [redacted at the request of the State of Maryland]; www.ballot-integrity.org/docs/SAIC_Report.pdf.
9. B. Williams, “Presentation to the US Election Assistance Commission,” presentation, 5 May 2004; www.eac.gov/assets/1/AssetManager/testimony%20brit%20williams%20kennesaw%20university%20public%20meeting%20may%205%202004.pdf.
10. D. Chaum, P. Ryan, and S. Schneider, “A Practical, Voter-Verifiable Election Scheme,” *Proc. 10th European Symp. Research in Computer Security* (ESORICS 05), 2005, pp. 118–139; www.pretavoter.com/publications/esorics05.pdf.
11. G. Howland Jr., “www.bigbrother.com,” *Seattle Weekly*, 9 Oct. 2006; www.seattleweekly.com/2004-05-19/news/www-bigbrother-gov.
12. T. Campbell, *Deliver the Vote: A History of Election Fraud, an American Political Tradition—1742–2004*, Carroll & Graf, 2004.
13. B. Schneier, “The Problem with Electronic Voting Machines,” *Schneier on Security*, 10 Nov. 2004; www.schneier.com/blog/archives/2004/11/the_problem_wit.html.
14. D.W. Jones and B. Simon, *Broken Ballots: Will Your Vote Count?*, Center for the Study of Language and Information, 2012.

HAL BERGHEL is a professor of computer science at the University of Nevada, Las Vegas and is an IEEE and ACM Fellow. Contact him at hlb@computer.org.

This article originally appeared in Computer, *vol. 49, no. 9, 2016.*

An Internet of Drones

Robert J. Hall • *AT&T Labs Research*

The safe operation of drones for commercial and public use presents communication and computational challenges. This article overviews these challenges and describes a prototype system (the Geocast Air Operations Framework, or GAOF) that addresses them using novel network and software architectures.

Drones, flying devices lacking a human pilot on-board, have attracted major public attention. Retailers would love to be able to deliver goods using drones to save the costs of trucks and drivers; people want to video themselves doing all sorts of athletic and adventuresome activities; and news agencies would like to send drones to capture video of traffic and other news situations, saving the costs of helicopters and pilots.

Today, both technological and legal factors restrict what can be achieved and what can be allowed safely. For example, the US Federal Aviation Administration (FAA) requires drones to operate within line-of-sight (LOS) of a pilot who's in control, and also requires drones to be registered.

In this article, I will briefly overview some of the opportunities available to improve public and commercial drone operation. I will also discuss a solution approach embodied in a research prototype, the *Geocast Air Operations Framework* (GAOF), I am working on in AT&T Laboratories Research. This prototype system has been implemented and tested using *simulated* drones; aerial field testing with real drones is being planned and will be conducted in accordance with the FAA guidelines. The underlying communications platform, the *AT&T Labs Geocast System*,[1-3] has been extensively field tested in other (non-drone) domains with Earth-bound assets, such as people and cars. The goal of the work is to demonstrate a path toward an improved system for the operation of drones, with the necessary secure command and control among all legitimate stakeholders, including drone operator, FAA, law enforcement, and private property owners and citizens. While today there are drones and drone capabilities that work well with one drone operating in an area using a good communication link, there will be increased challenges when there are tens or hundreds of drones in an area.

Note that some classes of drone use are beyond the scope of this discussion:

- *Military drones*. The US military has been operating drones for many years and are the acknowledged world experts in the field. However, its usage scenarios are quite different, and many of its technical approaches are out of scope for this discussion, because they have resources and authority that are unavailable (such as military frequency bands) or impractical (high-cost drone designs and components) to use in the public/commercial setting. Instead, we seek solutions whose costs are within reason for public and commercial users and which do not require access to resources unavailable to the public.
- *Non-compliant drones*. It will always be possible for someone to build and fly drones

Additional Information

The following notes provide a few technical details to supplement the main discussion:

- We have consistently seen single-link connectivity over ad hoc WiFi at ranges of 100–125 meters in an open field at the surface of the Earth, for example, when held by a person or mounted in a car. However, we expect link ranges for ground-to-air of hundreds of meters (400+), with imperfect connectivity even in excess of a kilometer. We expect air-to-air to exceed even these, though we of course plan to test to verify this. Electromagnetic waves propagate much better in free space than near the Earth. For slow-moving drones (tens of m/sec), these link ranges seem fine for collision avoidance.
- For example, airspace awareness among a cloud of drones operating near each other can be accomplished in $n \lg n$ messages,[1] rather than the n^2 messages required by traditional IP-addressed approaches. In addition to this algorithmic savings, the design of the SAGP wireless geocast protocol[2] does not rely on extra management packets, so spectrum that would normally be wasted on exchanging topology packets to establish routing tables in an IP-based wireless network is instead available to transport more information.
- The rectangle defined by the tuple <*OriginLat, OriginLon, length, width, orientation*> is constructed as follows. The origin lat/lon define a reference point as one corner of the rectangle. Draw a ray segment from there, of length *length*, having a bearing of *orientation* from true North. Next, draw a ray from the origin, of length *width*, having a bearing of *orientation* + $\pi/2$ from true North. Complete the rectangle defined by the origin and the tips of the two ray segments. Aligning the rectangle such that the first ray segment lies along the positive Cartesian y-axis, the interior of the rectangle comprises the points (0 .. *width*) × (0 .. *length*).

References

1. R.J. Hall, "A Geocast-Based Algorithm for a Field Common Operating Picture," *Proc. 2012 IEEE Military Comm. Conf.*, 2012; doi:10.1109/MILCOM.2012.6415848.
2. R.J. Hall, "An Improved Geocast for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 2, 2011, pp. 254–266.

that do not obey the protocols of our system. For example, we will not discuss *defense against drones*, such as electromagnetic pulse (EMP) weapons, jamming, or trained birds-of-prey.[4] However, we hope to work toward a framework for safe and secure large-scale drone use, analogous to establishing traffic laws for cars.

- *Drone application-layer issues.* Obviously, drones should actually do something useful once we have gone to the trouble to operate them safely. Often, this takes the form of capturing video or gathering other sensor data. This article does not address the issues involved in transferring large data sets from drone to ground or drone to cloud.

The rest of this article will give background on the communications system underlying the GAOF, the challenges of safe and scalable air operations, and how the GAOF addresses these challenges.

Background: Multi-Tier Geographic Addressing and the ALGS

It is natural to think of using commercial communications systems, such as the commercial cellular data system (4G/LTE or GSM), to connect drones to servers in the Internet or to users operating smart phones. The difficulty here is that coverage of the cell system is not universal: what happens to the drone when it flies into a gap in coverage? Of course, the link to a remote user will not work, but there may be necessary functions, such as collision avoidance and mutual awareness, that operate between the drone and other drones, which must function in order to avoid trouble.

On the other hand, it's attractive to use local wireless ad hoc networking[2] to connect drones to each other for purposes like awareness, collision avoidance, and zone declarations. (See the "Additional Information" sidebar for a discussion of link range.) The only problem with that is it does not address the need for efficient access to remote resources, such as a remote non-LOS controller, airspace controllers, or cloud servers that might provide necessary information like restricted-fly zones.

The AT&T Labs Geocast System (ALGS) combines these two ideas into a multi-tier internetwork[1] that allows packets to flow across either tier individually, or both concurrently, depending on availability of the tiers (for example, whether the drone is in cell coverage, or whether the drone has ad hoc WiFi capability) and on the source and destination of the packet. The ad hoc wireless tier uses WiFi devices in *ad hoc mode* (802.11 IBSS Mode); the long-range tier uses a cellular data link (LTE) to connect the drone to a georouter resident in the Internet. When a drone lacks the hardware necessary for one or the other of the two tiers, the ALGS software seamlessly falls back to single-tier use. Note, however, that the architecture does provide naturally for *relaying*. So, for example, if one drone has both tiers available, it can act as a *relay*, transferring long-range packets into the short-range tier (or vice versa), so that a single-tier drone can receive messages that come via the relay from sources on the other tier. This is useful in many scenarios; for example, one could orbit a two-tier-capable drone at higher altitude above an area of operations in a valley where cell coverage was nonexistent, allowing remote awareness and control of drones operating at lower altitude within the valley.

The second key concept in the ALGS is *geographic addressing*. In a geographic addressing (GA) network, packets are addressed to all devices in a specified subset of space. In ALGS, an address is a circle, defined by the latitude and longitude of its center point, and its radius in meters. In many field applications (like drone operations), one wishes to address messages to devices in a physical area, not really knowing in advance which devices are there. For example, to support an airspace-awareness display, one wishes to send a query to any and all devices in the area of interest, without knowing if any are there or, if so, their identities. That, after all, is the purpose of the query. By providing this geographic addressing and routing capability at the network level, it can be made as efficient as possible. It can also be provided as a shared service to all applications needing this capability, rather than having each service implement its own GA infrastructure, which would waste network bandwidth and other resources.

For packets that traverse the ad hoc wireless tier, GA also has the advantage of being fundamentally more efficient, due to exploiting one-to-many broadcasts at the physical layer. See the "Additional Information" sidebar for further discussion of the efficiency advantages.

In summary, the ALGS provides geographic addressing seamlessly over a multi-tier internetwork that allows both long-range communications and short-range, low-latency communications among drones in mutual proximity.

Challenge 1: Airspace Awareness

To fly a drone, an operator must know at all times where it is and what else is nearby that may conflict with it. In addition, it is reasonable to assume that law enforcement or the FAA may wish to establish some control over airspace around particular areas at particular times. Such controllers will need to know locations of all drones flying in the area as well as other information about them, such as velocity, altitude, flight plan, and system health. Finally, the onboard software controlling the drone itself needs to know positions and altitudes of nearby drones and other objects, in order to enable automated actions like avoiding collisions.

Airspace Awareness is the general problem of querying all devices in an area for a defined set of information, usually in order to populate and update an *operating picture* of the airspace. When the same picture must be present on all the devices, such as when all drones in an area are attempting to avoid colliding with each other, we term it a *common operating picture*.

Airspace awareness is challenging for at least two reasons. First, the problem of every device telling every other device its status information sounds at first blush to require at least $n(n - 1)$ messages per unit time. Such quadratic growth could swamp a wireless network if the number of drones in an area increased sufficiently. Second, the operating picture changes quickly, because devices move in and out of the area, land and take off, run out of charge, turn on, or purposely sleep their communications for defined periods to conserve power. Thus, it is important not only to keep updating the picture regularly, it is also critical to keep track of how old the information in the picture is; knowing only the position of a nearby drone as of five minutes ago is of limited use in collision avoidance.

Figure 1 shows a simulated awareness display from a GAOF smart phone app. Green dots show positions of drones, with a small data display for each showing ID and velocity. The colored "tails" show movement history, with color coding indicating recency of the information: green indicates the past 15 seconds, yellow 15–60 seconds in the past, orange 60–240 seconds old, and so on. The red rectangle is a no-fly zone that has been declared in the area using the app. The app uses the Field Common Operating Picture (FCOP) protocol[3] to obtain the data needed to populate the display. Two commanded waypoints are marked as well; onboard augmentative control software has planned a route between them that avoided crossing the no-fly zone.

GAOF provides a general awareness framework based upon the FCOP awareness protocol,[3] which is a distributed algorithm for a common operating picture. Basically, each device wishing to monitor an area formulates a query that describes the desired information and periodically transmits this query in a geographically addressed packet to the area of interest. The devices in the area, under control of the FCOP protocol, respond back to the area surrounding the location of the device issuing the query with response messages containing their information. GAOF's use of FCOP has a couple of advantages. First, when querying into the ad hoc wireless tier, the protocol reduces the message complexity to $O(n \lg n)$ average case per unit time, which is a major scalability advantage. Second, by responding back to an area (rather than to only the initial querying device), the algorithm further reduces message complexity by having devices listen to all responses, not just to responses to its own queries. Basically, devices know that if they have responded to a query recently to an area containing a different querying device, they need not respond again so soon. How soon is “soon” depends upon a GAOF system parameter whose value should be guided by expected typical velocities of drones in an area.

Challenge 2: Non-Line-of-Sight Control

The goal of non-line-of-sight (NLOS) control is to allow a drone controller to be physically located arbitrarily far from the drone itself. The military flies drones this way, for example. While the FAA’s rules do not permit non-line-of-sight operation today, it is useful to consider how it could be achieved if it becomes a permitted use. The primary challenges of NLOS control are latency, security, and routing. Long delays, either in moving awareness information to the controller or in moving commands from controller to drone, could lead to loss of control, collisions, or crashing into terrain. Security is critical, since one does not want an unauthorized operator to seize control of the drone via a network connection. Routing is a challenge whenever the drone is in an area with spotty or nonexistent coverage; clearly, loss of contact is a significant control challenge.

GAOF addresses the latency challenge using multiple techniques. First, we assume a *WayPoint-controlled Drone* (WPDrone). That is, control of the drone is communicated by sending it high-level commands, such as a target lat/long/altitude, a hover (hold) command, or a landing command. We do not attempt low-level control, such as manipulating rotor speeds or control surface pitches. This eliminates the need for millisecond-level latency, with onboard control software dealing with that level. This does not entirely solve the problem, of course, as things can come up quickly, especially if another drone approaches. Collision avoidance is discussed below.

With the technology discussed here, we would secure the drone primarily using standard cryptographic techniques. Each command is securely signed by the controller using a secure session key shared with the onboard GAOF software, and each command must pass the signature test on reception before it is presented for execution. If privacy is desired on this link, an additional layer of encryption will be added to obscure the waypoints.

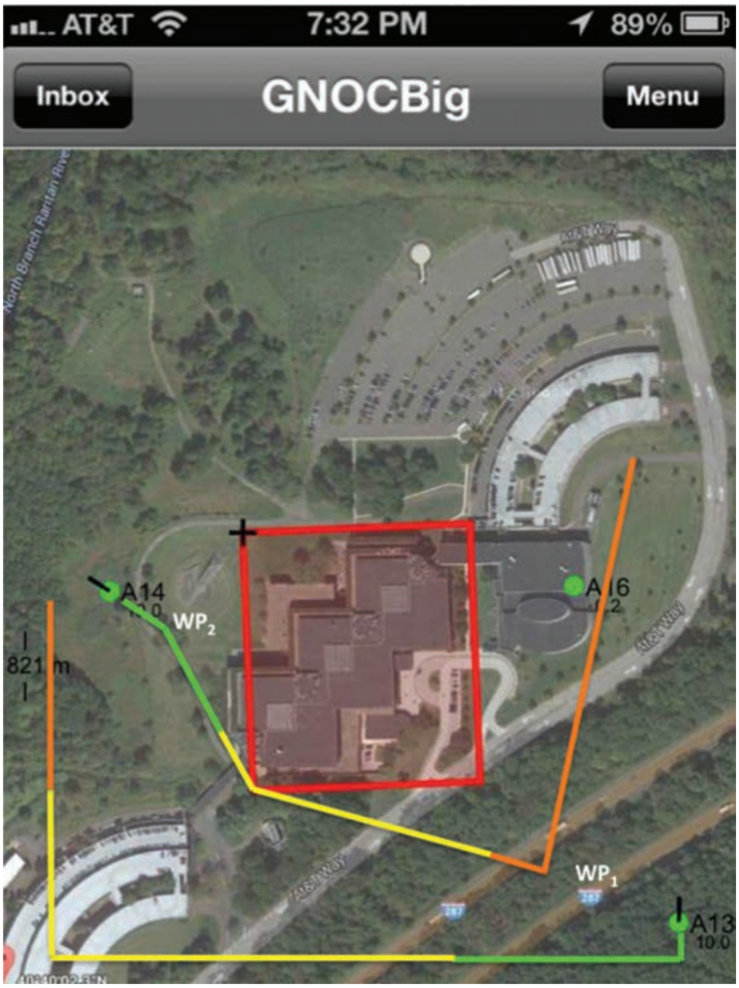


Figure 1. Airspace awareness display from a GAOF smart phone app showing three (simulated) drones (green dots) and a no-fly zone (red rectangle). Multi-colored lines are position histories of the two moving drones, with color coding for recency in time, green being most recent information. The Augmentative Control module has planned a path around the red zone to replace the segment from waypoint WP_1 to WP_2. (Image and map data © 2014 DigitalGlobe, Google, USDA Farm Service Agency.)

To route the commands, GAOF uses an enhanced message-addressing primitive termed a *logicast*. In brief, a logicast is a geographically addressed message that is further filtered by a logical predicate. That is, the address of the message is all devices in the area on which the predicate evaluates to true. In the case of NLOS control commands, the logical predicate is true iff the device’s unique ID matches that in the predicate. This allows the GA system to do the heavy lifting of finding out how to get the packet to the drone, while avoiding delivering the packet to all the drones in the area.

Challenge 3: Augmentative Control

There will always be restrictions on where drones can fly and when. We therefore need a way to *augment* the control of drones by additional commands when appropriate and authorized. These augmentations are applied to modify, cancel, or replace commands received from the controller. They should be in force even when the drone is not within cell coverage or, indeed, when it may not be within range of *any* communication network; consider, for example, restrictions that would keep drones out of active flight paths near major airports. Major airports may experience overloaded cell systems during prime travel hours, so drones may have only limited network connectivity in the area.

Augmentative Control (AC) is challenging in several ways. First, it must be secure; only parties having authorized signing keys can define airspace restrictions or other augmentative control commands. Second, drones may not always be assumed to have an Internet connection such that they can constantly keep up with a remote server that may be issuing augmentative control commands. In fact, it must be possible to issue new commands to a drone in flight, even if the flight area is not within cell coverage. For example, a law enforcement officer in the area near the drone may wish to command it to land due to an emergency situation arising, where the emergency has caused terrestrial communications systems to be unavailable. Third, there must be a clear policy hierarchy of AC commands: what if, for example, a collision avoidance AC module commands a course change into a red zone?

In GAOF, the primary augmentative control mechanism is the *zone*. A zone is a rectangular area defined by the tuple of <*OriginLat, OriginLon, length, width, orientation*>. See the "Additional Information" sidebar for how this defines a rectangle on the surface of the Earth. In GAOF, all zones extend to all altitudes, though in the future, a richer 3D representation may be developed. There are currently two types of zone:

- A *red zone* denotes a *no-fly zone*; drones must not fly into or within red zones.
- A *green zone* is a *geofence*; once inside, a drone must not leave a green zone.

Figure 1 shows a red zone that diverted the flight plan of a drone to avoid it flying over some buildings.

Zones are communicated to drones via a distributed agreement protocol. Any device can send a *zone declaration*, which tells devices the IDs of all zones currently in force nearby. If a device hears such a message and finds within it a zone declaration of which it was unaware, it issues a *zone information request* to the device, which replies with the information. Through this protocol, the drones in an area eventually come to agree on the current set of zones. A zone deletion is communicated via the same mechanism. When a drone receives a new zone definition, it checks the digital signature of the zone to ensure that it is authorized. In this way, all devices can pass around zone information, but only authorized key holders can create or delete zones.

If a GAOF-style platform were to be adopted and deployed in the future on a wide scale, the zone mechanism would be expressive enough to enable these AC example scenarios:

- Airport authorities define a no-fly zone covering sensitive areas in and around the airport;
- Drone operator defines a containing geofence to ensure the drone won't venture beyond property lines;
- Secret Service creates a temporary no-fly zone over the path of a presidential motorcade route; and
- Law enforcement creates a no-fly zone around a particular drone to force it to land, subsequently deleting the zone once it has served its purpose.

The Zones AC component is just one of the AC modules deployed into the GAOF onboard software control architecture. The detailed software architecture of GAOF is beyond the scope of this article, but it's based on a blackboard architecture, where AC modules make suggestions of new flight plans to a single DronePilot module that decides among them based upon a policy definition.

The Zones AC module monitors the position and current flight plan of the drone. If it detects the drone is about to leave a green zone, it suggests a hold/hover several meters before the boundary. If it detects that the ray segment from its current position to the next waypoint would cross into a red zone, it uses a graph algorithm to suggest a modification to the flight plan to reach the waypoint but skirting around the red zones. (See Figure 1 for an example of this.) If no such suggestion exists, it instead suggests a stop before crossing the red zone boundary.

AC mechanisms other than zones are useful as well, and can be incorporated into GAOF's onboard software architecture. For example, one mechanism detects when a drone is about to crash due to low battery and commands a controlled landing instead; another implements the FAA's drone operation ceiling, commanding the drone to descend if it gets above the limit. Many others are, of course, possible.

Challenge 4: Collision Avoidance

An important special case of augmentative control is avoidance of collisions.

This is made possible by the drones collecting a common operating picture containing positions, altitudes, and velocities of objects around the drone. If a collision condition is imminent, perhaps the simplest AC action to suggest (with high priority) is a hover action. Drones incapable of hovering would have to have more complex responses, which are not yet implemented within GAOF. Collision-avoidance systems in fixed-wing aircraft, such as the Traffic Alert and Collision Avoidance System (TCAS),[5] have a long and relevant history.

It is entirely possible to include GAOF components on non-drone objects. Analogous to those flashing red lights on buildings and transmitter towers, such beacons would participate in the operating picture by sending out their positions and possibly zone definitions so that drones could avoid crashing into buildings, towers, bridges, and so on.

It is tempting to believe that "the sky is big, and drones are small, so why bother with collision avoidance?" One response is that buildings and other large obstacles are hazards to drones and should be addressed in some way, such as by putting active zone-declaring beacons on them, as previously discussed. More subtly, if a drone plans its way around the edges of no-fly zones, this tends to concentrate a large cloud of drones onto near-one-dimensional lines, which greatly increases the likelihood of interference, a phenomenon first documented in the context of commercial aviation.

Collision avoidance must work in proximity to a drone even if the area is not within cell coverage. This is a key advantage of the ALGS-based multitier architecture, as an ad hoc WiFi-capable drone can communicate locally even in such conditions. ALGS and FCOP have been tested successfully in ad hoc-only field conditions to provide a near real-time operating picture to ground units. Local messaging also has lower latency, which is an advantage in collision avoidance as well.

A future of safe, secure, and reliable large-scale public drone operations promises to bring many benefits to society, but to achieve this, various challenges must be overcome. This article has given a brief tour of four such challenges: airspace awareness, non-LOS control, augmentative control, and collision avoidance. The Geocast Air Operations Framework, based on a multi-tier geographic addressing internetwork for communications, is one option to address these challenges.

Acknowledgments

The views expressed in this article are solely those of the author and do not necessarily reflect the views of AT&T, IEEE, or any other entity.

References

1. R.J. Hall et al., "Scaling Up a Geographic Addressing System," *Proc. 2013 IEEE Military Comm. Conf.*, 2013, pp. 143–149.
2. R.J. Hall, "An Improved Geocast for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 2, 2011, pp. 254–266.
3. R.J. Hall, "A Geocast-Based Algorithm for a Field Common Operating Picture," *Proc. 2012 IEEE Military Comm. Conf.*, 2012; doi:10.1109/MILCOM.2012.6415848.
4. E. Ackerman, "Dutch Police Training Eagles to Take Down Drones," *IEEE Spectrum*, 1 Feb. 2016; http://spectrum.ieee.org/automaton/robotics/aerial-robots/dutch-police-training-eagles-to-take-down-drones.
5. J.K. Kuchar and A.C.Drumm, "The Traffic Alert and Collision Avoidance System," *Lincoln Labs J.*, vol. 16, no. 2, 2007, pp. 277–296.

Robert J. Hall is a principal investigator at AT&T Laboratories Research. His work focuses in the areas of automated software engineering, requirements engineering, modeling and simulation, scalable wireless network protocols, and cloud performance engineering. Hall has a PhD in electrical engineering and computer science from the Massachusetts Institute of Technology. He is a Fellow of Automated Software Engineering and member of the Steering Committee of the IEEE/ACM International Conferences on Automated Software Engineering. He serves as Editor in Chief of *Automated Software Engineering*, an international journal, and is an ACM Distinguished Scientist. Contact him at hall@research.att.com.

This article originally appeared in IEEE Internet Computing*, vol. 20, no. 3, 2016.*

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*

Ten Open Questions for Techno-Optimists

Shane Greenstein
Harvard Business School

From what I can gather from several recent articles, many serious pundits have arguments with the views of "techno-optimists." A techno-optimist appears to be somebody who has blind faith in the power of technology to cure all ills and, in particular, to create economic growth.

I understand skepticism with an optimist, and asking someone naive to "show me the money." As a rule of thumb, it is a good idea to be skeptical of bloggers and futurists who lack hard economic statistics.

That said, skepticism is an attitude, not a conclusion. Go ahead and ask for hard data, but do not close an open question. Sometimes, innovative IT has had a spectacular impact on the economy, and sometimes it has not. It takes work to recognize the difference. Indeed, there is a set of norms and a range of standard tools for settling these questions—such as quality-adjusted price indices, revenue adjusted for inflation, and a range of economic accounting techniques.

This column (shamelessly) aims to educate journalists about some of the big open questions in research that studies productivity growth from innovative IT. While I'm at it, I hope the topic informs graduate students about great topics for a thesis. Be forewarned, however: it is a little shocking how much (or how little) we really know about recent experience. Many of these questions are wickedly difficult. The only thing I can promise with certainty is the absence of an easy answer.

Why Are Online Pictures and Videos Everywhere Except in the Productivity Statistics?

Tim Berners-Lee invented the Web to share graphs and pictures and other media, and nobody paid anything for licenses for the software. It spread everywhere, so we got a combination of low cost and wide use. Seems good for economic growth, right? To be sure, a space alien might see the Web for the first time and conclude that it was invented primarily to share baby pictures and cat videos, and to illustrate men and women in, um, acts of procreation. Reflect for a thoughtful moment and really look beyond the superficial. There are millions of webpages. An enormous ecosystem of browsers and server software and networking infrastructure supports its use, and some of these actors—mainly ISPs—make some revenue, well north of $50 billion in wireline access. Online advertising is a little less. Is that it? How about online sales, which is less than 8 percent of US sales revenues, but does not make large profits? What boost did the US get—a one-time surge in growth or a steady set of gains until now?

How Much Productivity Does Email Produce?

Email has a different history than the Web, but its rise motivates almost the same types of questions. Nobody ever paid a licensing fee to make use of this invention. Last year, more than 200 billion emails were sent per day. To be sure, 75 percent of those emails are spam, but that still leaves a lot of quality communication. Where do we see those gains? Gmail and Hotmail, the two largest email providers, charge nothing. Microsoft's Outlook makes revenue, but that is not the contribution of email to GDP. Surely not. So, how much does email contribute? When did email lead to growth—20 years ago, when it largely displaced postal mail of letters? How about recently?

What Were the Gains from Reduction in Search Costs?

Who among you uses portals, search engines, recommendation sites, and online maps? Okay, everyone can put down their hands. Remember how your parents used to fumble through the Yellow Pages? Remember how a cross-city trip involved laying out a great big paper map on the kitchen table? Remember when a cousin's recommendation determined which restaurant in a new neighborhood you visited? Difficult as it might be for teenagers today to comprehend, everyone did manage in those dark ages. Still, today is much better. How much did the reduction in search costs bring to GDP, if anything, and when? Advertising supports some of this activity, which surely is only one aspect of a broad

 2469-7087/16/$33.00

This article originally appeared in IEEE Micro, *vol. 36, no. 4, 2016.*

economic gain. Well, how do you measure those gains?

What Were the Gains from Making the Long Tail Available?

Ebay, Amazon, Craigslist, and a gazillion sites made the long tail available—of books, music, clothing, memorabilia, and millions of other products. What an enormous variety lies at everyone's fingertips. What does the long tail contribute to economic growth? There have been some estimates in specific product categories, such as books and shoes, but none across the entire economy. Were most of those gains realized in the late 1990s? Do those gains today simply grow along with Amazon's growth? Are most of the gains not realized as revenue and, therefore, not measured?

Up-to-Date Online News Is Additive. Is It Productive, Too?

The creation of online media ushered in an era for news junkies. Again, it would be hard for a child to comprehend how anybody checked sports scores or learned timely financial news (without access to expensive Bloomberg terminals), but we did manage before. More deeply, what is the contribution of more timely information to economic productivity? It seems like many gains are not measured. If they are, where do those gains show up in national statistics—and in which industries?

Did the Rise of Remote Work Change Productivity?

The cell phone and smartphone made work more mobile. The deployment of cloud is pushing in the same direction. There have been a few studies of a consumer's willingness to pay for mobility, but very little about its effect on work. How did the economy gain from the introduction of mobility into work life? Just like the other questions: Some of the gains might show up in productivity statistics, or in prices, or in the hours worked, or in labor participation rates—or not. How do we know?

How Much Did Wikipedia Benefit the Economy?

Founded in 2001, Wikipedia is the third largest repository of human knowledge in the world today, exceeded only by the British Library and the Library of Congress. It is a top-20 site in every developed country. Traditional GDP measurement would value the advertising, but Wikipedia does not have any. Is Wikipedia's value the displaced revenue at Encarta, Britannica, World Book, and Colliers? Or is Wikipedia's worth the value of the time put in by all its volunteers? Or is it the value to users, and the time savings it affords? What concept of value is appropriate, and how would you implement it and measure its growth?

Enterprises Do Not Own All Their IT. Does That Mean They Are More Productive?

We are well past the days when an organization owned all its IT. Today we live in the era of cloud, networked services, rented manufacturing facilities, and marketing with social graphs. We are far past the era when a firm bought inputs and produced outputs in a manufacturing process and its productivity could be measured inside the manufacturing plant. In the extreme examples, firms today such as Uber and Lyft own few cars and employ few people, yet the firms are worth tens of billions. Has more value been created for GDP, if any, by these new organizational forms that leverage external resources? How would you know?

How Big Were the Gains from Serving Low-Density Areas?

Improvements in Internet access led to gains in GDP in places that had limited access to retail outlets. Traditional price indices have an urban bias, because the information is easier to collect in urban stores. Rural America comprises the experience of 15 percent of the US population (45 million people), and a much higher percentage (and number of people) in the developing world. Again, how big were the unmeasured gains? Were most of the gains realized during the dot-com boom, or has there been steady progress as the Internet has gotten faster?

What Is the Value of the Creative Commons License?

A little less than 20 years ago, some legal scholars invented the licenses known as the creative commons license. It is designed to permit sharing of copyrighted material in much the same way that open source has licenses for sharing code. Millions of websites operate with this clever legal adaptation; it is integral to some popular sites with user-provided content, such as YouTube. What is the value of this invention? Do the price indices properly capture the extraordinary decline in the cost of sharing photography and video recordings? If not, how would you make a more accurate index?

These questions presume that many economic gains from the deployment of the commercial Internet went largely unmeasured. Blame flaws in traditional price indices, inadequate definitions of revenue, and flawed productivity attribution exercises.

These questions also tend to imply that the processes that created growth in the recent past will continue to create growth in the future. That motivates the research question. Although it is good to fix recent growth statistics, it is even better to measure the gains in the near future.

If these two features make me an optimist, then call me that. More to the point, if you're an economic researcher and you do not like this list of 10 questions, then make your own. Let's focus on one of the biggest unaddressed economic topics of our time. MICRO

Shane Greenstein is a professor at the Harvard Business School. Contact him at sgreenstein@hbs.edu.

Editor: **Jane Cleland-Huang**
DePaul University
jhuang@cs.depaul.edu

Monitoring Requirements in Systems of Systems

Michael Vierhauser, Rick Rabiser, and Paul Grünbacher

LARGE-SCALE, heterogeneous software systems are ubiquitous in many domains. Often, such systems are part of systems of systems (SoSs) working together to fulfill common goals resulting from domain or customer requirements.[1] SoSs comprise hardware and software systems that are frequently provided by different vendors following their own development strategies and release plans.[2] SoS engineering has been compared to developing cities in contrast to constructing buildings.[3] SoSs typically aren't planned and engineered top-down but come into existence gradually, often as the result of decades of development by independent teams.

Consider an example from the automation software domain. In a steel plant, heterogeneous software systems interoperate to automate and optimize metallurgical tasks. Systems developed by different teams and even vendors interact to automate processing iron ore; melting and refining ore to produce steel; casting liquid steel into solid slabs; and producing girders, steel sheets, and other products. A steel plant SoS must coordinate material transportation, quality control, and logistics during production, and it controls hardware such as torch-cutting machines, transport vehicles, or robots. The correct interplay between these software and hardware systems is crucial to guarantee continuous, uninterrupted production and high-quality products.

So, it's essential to ensure that an SoS complies with its requirements. SoS requirements are owned by diverse stakeholders and exist at different levels, for different systems, and in different artifacts. They often overlap, cross-cut, or even conflict, and allocating them to specific systems or components isn't always possible. Furthermore, requirements stem from physical processes, resulting in the need to carefully consider the dependencies between software and hardware.

Also, an SoS's full behavior emerges only at runtime, when the involved systems interact with hardware, third-party systems, or legacy systems. Thus, SoS engineers must monitor if and how the SoS meets its requirements at runtime[4]—for instance, to verify components' correct timing or measure performance and resource consumption. Monitoring can happen continuously at runtime, to give instant feedback on behavior violations, or post hoc, on the basis of recorded event traces and data logs.

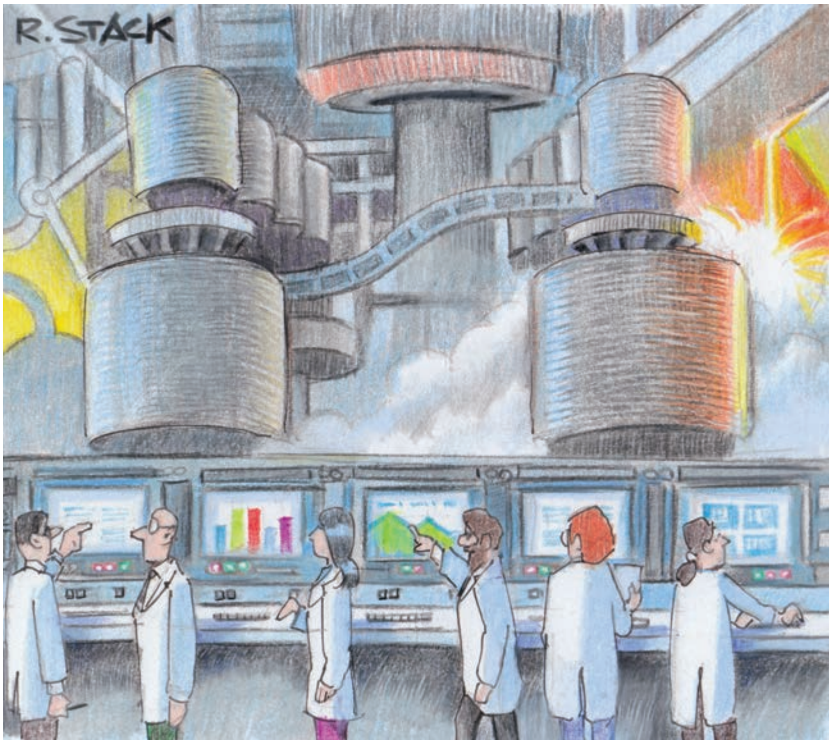
The Challenges of SoS Monitoring

As Neil Maiden pointed out previously in this magazine department, "Software systems generate various events, which can provide data that can be used to test for requirements compliance."[5] SoS en-

 2469-7087/16/$33.00 © 2016 IEEE

gineers need tools that capture these events and data across the SoS to monitor the requirements.

For example, the hardware automation layer in a steel plant SoS provides sensor data and machine information such as the blast furnace temperature, the speed of the strand moving in the casting machine, or the pressure of the nozzles spraying water on the cast steel. So, optimizing steel quality relies on data on the strand's speed provided by sensors in the rollers pushing it forward, data on the temperature provided by sensors in the mold, and information about the material mixture from analysis systems. Collecting and interpreting this data is vital, for example, when calculating the optimal amount of water to spray to cool the steel.



The complexity and heterogeneity of SoSs complicates runtime-monitoring requirements. As our steel plant example demonstrates, monitoring must occur at different layers and levels of granularity, and it relies on collecting different kinds of events and data from multiple sources. For instance, retrieving information about intersystem communication in an SoS requires instrumenting the diverse system interfaces. Checking interprocess communication requires analyzing the interactions between the involved processes. So, monitoring must work across systems because SoS requirements can't be allocated to single systems or components, as we mentioned before. Instead, properties must be checked across the boundaries of multiple constituent systems.

This problem goes hand in hand with instrumenting heterogeneous, domain-specific technologies in architecturally diverse systems. Whereas developers can, for example, use techniques such as aspect-oriented programming to instrument mainstream programming languages without manually modifying code, instrumentation for legacy systems is much harder. Also, different types of requirements must be checked at runtime, including global invariants, range checks of variables, temporal constraints on events, or architectural rules constraining component interactions. Finally, SoSs exist in many different versions and variants owing to the continuous, independent evolution of their constituent systems. Adaption and reconfiguration of monitoring solutions thus becomes essential.

ReMinds

In response to these challenges, and on the basis of industrial monitoring scenarios, we've been developing the ReMinds (Requirements Monitoring Infrastructure for Diagnosing SoSs) tool (see Figure 1).[6] Engineers can use it to instrument systems in an SoS to extract events and data at runtime and to define requirements as constraints to check expected behavior and properties. ReMinds can also visualize and explain requirements violations to facilitate diagnosis.

ReMinds has been applied to monitor requirements at different layers of a steel plant automation SoS. During continuous monitoring, events and data from different systems are collected at runtime and checked immediately to determine requirements violations. This lets engineers inspect certain components' timing on the fly. For example, the optimization system for continuous casting must finish an optimization calculation x seconds after the steel plant operator or another system in the SoS triggers that calculation.

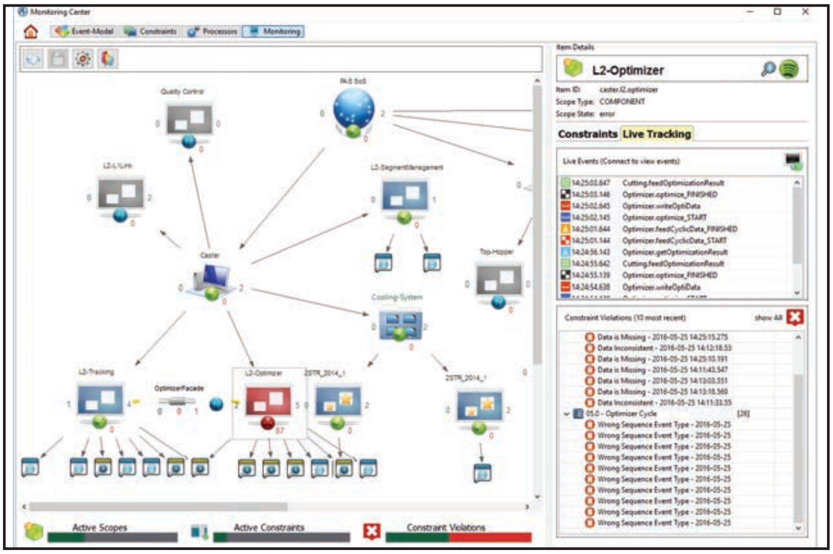


FIGURE 1. The ReMinds tool supports requirements monitoring in systems of systems (SoSs). Engineers can instrument systems in an SoS to extract events and data at runtime and to define requirements as constraints to check expected behavior and properties. ReMinds can also visualize and explain requirements violations.

Checking this behavior requires instrumenting several systems in the SoS, such as the optimization system and the human–machine interface, which have been developed using Java and .NET. For this purpose, ReMinds supports cross-system and cross-technology instrumentation and analysis of events and data.

When online monitoring isn't possible—for example, when a problem emerges at runtime for which engineers haven't yet defined a constraint—ReMinds supports capture and replay. Engineers can use captured event traces from the system and replay them using their existing simulators to reenact what happened in the SoS. ReMinds also supports capture-and-compare analyses of recorded event traces, which are particularly useful to support evolution. Specifically, industrial SoS components are frequently upgraded—for example, during plant modernization. Such selective upgrades can have unforeseen effects on the SoS's overall behavior. ReMinds thus can capture event traces before and after an upgrade and compare them to compute and visualize differences in behavior.

The shift from statically defined requirements toward requirements as runtime entities—for example, as in current cyberphysical production systems—will continue. So, requirements will increasingly become ubiquitous throughout the software life cycle. Consequently, the development of solutions to monitor requirements of SoSs in different domains will become essential, providing many interesting challenges for both practitioners and researchers.

References

1. M.W. Maier, "Architecting Principles for Systems-of-Systems," *Systems Eng.*, vol. 1, no. 4, 1998, pp. 267–284.
2. C.B. Nielsen et al., "Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions," *ACM Computing Surveys*, vol. 48, no. 2, 2015, article 18.
3. L. Northrop et al., *Ultra-Large-Scale Systems: The Software Challenge of the Future*, Software Eng. Inst., Carnegie Mellon Univ., 2006; www.sei.cmu.edu/library/assets/uls_book20062.pdf.
4. W.N. Robinson, "A Requirements Monitoring Framework for Enterprise Systems," *Requirements Eng.*, vol. 11, no. 1, 2006, pp. 17–41.
5. N. Maiden, "Monitoring Our Requirements," *IEEE Software*, vol. 30, no. 1, 2013, pp. 16–17.
6. M. Vierhauser et al., "ReMinds: A Flexible Runtime Monitoring Framework for Systems of Systems," *J. Systems and Software*, vol. 112, 2016, pp. 123–136.

MICHAEL VIERHAUSER is a researcher at the Christian Doppler Laboratory for Monitoring and Evolution of Very-Large-Scale Software Systems at Johannes Kepler University Linz. Contact him at michael.vierhauser@jku.at.

RICK RABISER is a senior researcher at the Christian Doppler Laboratory for Monitoring and Evolution of Very-Large-Scale Software Systems at Johannes Kepler University Linz. Contact him at rick.rabiser@jku.at.

PAUL GRÜNBACHER is the head of the Christian Doppler Laboratory for Monitoring and Evolution of Very-Large-Scale Software Systems and an associate professor at Johannes Kepler University Linz. Contact him at paul.gruenbacher@jku.at.

This article originally appeared in IEEE Software*, vol. 33, no. 5, 2016.*

Displays as a Material: A Route to Making Displays More Pervasive

David Sweeney, Nicholas Chen, Steve Hodges, and Tobias Grosse-Puppendahl, *Microsoft Research*

The plummeting cost of TVs and computer monitors, widespread adoption of mobile devices, and advent of wearable devices means that there are now more digital displays than ever before. But these are still outnumbered by the vast array of nondigital displays in the world in the form of text and graphics permanently printed on various objects around us. Turning these static surfaces into dynamic displays offers several benefits, including improved awareness of and faster access to relevant information;[1] more freedom in terms of display locations, which would improve ergonomics and aesthetics; and the potential for new, more personalized products with ever-changing designs that can seamlessly blend into the surroundings.

Today's displays are typically thought of as discrete components or modules around which devices are constructed. For digital displays to become more pervasive, they need to have the properties of materials like textiles and plastic film and other construction materials. Such an approach unlocks many scenarios currently restricted to static, printed displays. For example, a digital display material that can be cut and stretched could be applied to or built into arbitrarily shaped objects and could scale from a device- to building-size display.[2,3]

In our research, we're working to realize this kind of display material. Fundamental to our approach is a move away from the row-column addressing architecture that dominates traditional displays. More specifically, we propose an architecture that relies on *autonomous pixels*—that is, pixels that independently sense input and convert it to a corresponding visual output. Two different prototypes reveal the challenges and potential of autonomous pixels, highlighting how digital displays supplied as a flexible material can help foster the development of new applications.

A seamless high-resolution display for large-scale applications, such as billboards, is all but impossible with a grid topology.

LIMITATIONS OF THE GRID

A major barrier that has limited the versatility of contemporary displays comes from the hegemony of the grid-based topology used in their construction. This architecture, in which pixels lie at the intersection of a row wire and a column wire, provides a straightforward and convenient means for addressing each pixel. The grid topology is ideal for creating displays with rectangular shapes, because wires can simply extend across the full area of the display. But nonrectangular displays are difficult and therefore more costly to manufacture, because they need exotic wire-routing schemes, and, even then, the shapes they can take are extremely constrained.

Realizing a nonflat matrix display presents an even greater challenge. The fragility of the fine row and column wires in a grid-based display—where a single break can cause numerous pixels to stop working—means that these displays rarely support bending, compression, or stretching. But if these limitations could be overcome, it would unlock many new applications for pervasive digital displays.

Another limitation that arises from the grid topology is scalability. Increasing the size of a single display is only feasible within limits; ultimately, the number and density of wires becomes unworkable with a very large display. The alternative is to tile smaller displays, but modern active matrix display panels have an inherent bezel that precludes seamless tiling. The bezel is necessary because the row and column wires are connected to dedicated driver chips mounted along the outside edges of the panel. These chips are usually made using silicon, further introducing constraints on flexibility and fragility. As such, a seamless high-resolution display for large-scale applications, such as billboards and buildings, is all but impossible with a grid topology.

SPOTLIGHT

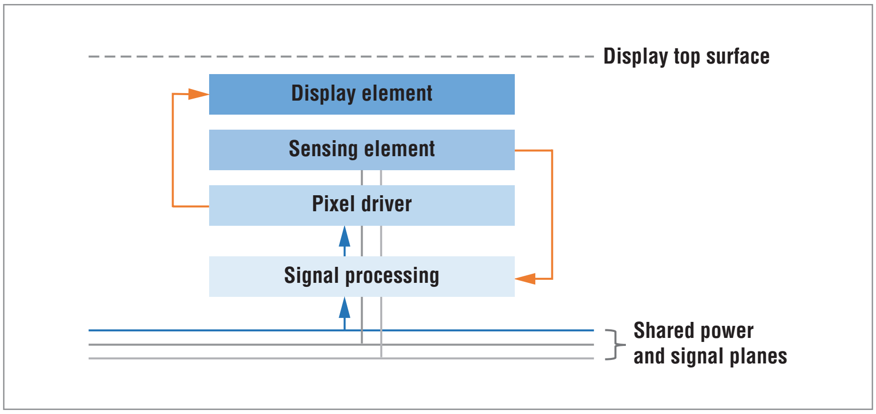


Figure 1. An autonomous pixel incorporates a sensing element that directly controls the pixel's visible contents. This shows the structure of our second prototype, where the photo sensor is directly behind the display media and faces up through it.

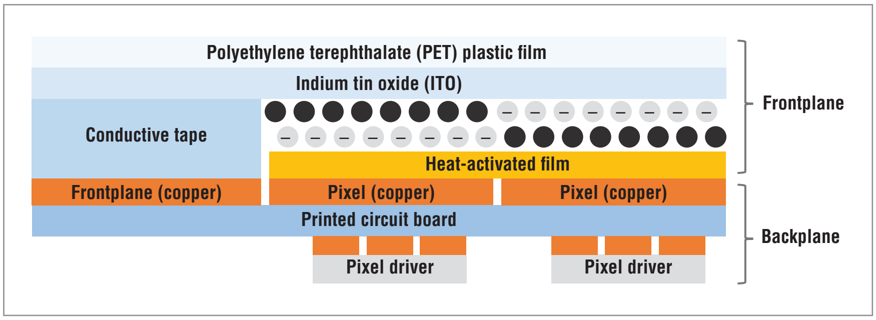


Figure 2. The stack-up in our prototype displays comprises multiple layers that are laminated together.

DISPLAYS USING AUTONOMOUS PIXELS

Departing from the grid opens up opportunities for more flexible, versatile, and robust displays. In previous work, Angie Chandler and her colleagues demonstrated that it was possible to use serially connected display elements to create displays with a wide variety of shapes.[4] We use a different approach to create the same versatility.

In our architecture, the pixels are independently responsible for sensing a signal, processing that signal, and using this information to control the display. In this autonomous pixel scheme, the only electrical connections to pixels are via conductive planes, which deliver power and one or two global signals shared across all pixels (see Figure 1). By incorporating sensing and signal processing capabilities into each pixel site, this architecture opens up new ways to address pixels and transfer data, freeing the display from the confines of the row-column grid.

The autonomous pixel architecture offers several benefits over a display built using a row-column matrix. Chief among these is that the display uses conductive planes instead of a delicate grid of wires: without dedicated signals running to each pixel, it becomes possible to cut out or shape parts of the display to support a variety of nonplanar applications. Also, autonomous pixel displays remove the constraint that pixels must be arranged in a rectangular grid; they can have differing shapes, arbitrary positions, and varying density. Finally, because the operating circuitry is local to each pixel, there's no need to interface with external driving electronics, making the displays seamlessly scalable.

Developing Prototype Displays

The detailed construction of a display depends on the underlying display technology, but most displays are based on a stack-up of different materials, which are laminated on top of each other. At a coarse granularity, there are two major layers—the frontplane forms the image from an optical perspective, while the backplane contains the electronics necessary to control the frontplane. The high-resolution, high-framerate, defect-free displays used for consumer devices, such as smartphones and TVs, are manufactured on sophisticated and expensive production lines. But if you're prepared to compromise on display quality, it can be remarkably straightforward to prototype custom displays in a lab environment.

The prototype displays we describe here are based on an electrophoretic display (EPD) frontplane—the same kind of technology used in e-readers like the Amazon Kindle.[5] This material is supplied in sheet form with the EPD media itself laminated onto a polyethylene terephthalate (PET) plastic film.[6] There is a transparent layer of conductive indium tin oxide (ITO) on the PET film, which is in contact with one side of the e-paper media, whereas the other side of the media is supplied coated in a layer of heat-activated film. This layer stack-up is shown in Figure 2.

The EPD media is a layer of tiny black and white spheres. The white spheres carry an electric charge, which means that by generating a suitable electric field, they can be moved toward or away from the display surface, thereby making the image more white or black, respectively. Such a field is created by applying a voltage

(a) (b)

Figure 3. Two macroscale, prototype displays demonstrating the viability of the autonomous pixel architecture. (a) Our first prototype comprises around 600 autonomous pixels with light-sensitive elements facing toward the rear. The display can sense an image on a computer screen behind and take a copy—a bit like photocopying without a copier. (b) Our second prototype features approximately 300 pixels with light-sensitive elements that face through the display surface, enabling an image projected onto the surface to be captured.

between the transparent ITO layer at the front of the display and the substrate to which the display is laminated. The field is controlled on a per-pixel basis.

The two macroscale prototype displays we developed demonstrate the viability of the autonomous pixel architecture using EPD material as the frontplane media. Each pixel combines a photodiode light sensor with EPD driving the circuitry in the backplane. The backplane substrate is a very thin printed circuit board, resulting in a flexible display that looks a bit like a large Polaroid print.

Our first prototype is the simpler of the two. It has the photodiodes behind the display, allowing the display to capture imagery from an emissive screen (see Figure 3a). In the second prototype, the photodiodes face forward, looking though the EPD material and sensing the intensity of light incident on the display (Figure 3b). This enables the display to be controlled by projecting light onto it. (For more detailed information about how we built our prototypes, see the "Step-by-Step Guide: Building a Display in the Lab" sidebar.)

Prototype Operation

The display senses light intensity at each pixel position using a photodiode. In the simpler of our two prototypes, if the intensity is above a certain threshold, the driving circuitry is activated, which causes the EPD media to change color. In our more advanced prototype, we add global "gain" and "threshold" signals. These two additional signals feed into the signal conditioning circuitry within each pixel and specify sensitivity and switching threshold. Erasing the display is accomplished by reversing the polarity of the power planes.

The design choices we made in our current prototypes aimed to keep implementation particularly simple. The bi-stable properties of the EPD media maintain state, bypassing the need to include memory elements at each pixel. The use of light conveniently allows us to leverage off-the-shelf displays and projectors to control the display. However, we imagine that with more engineering resources, our autonomous pixel architecture could be extended in a variety of ways.

In the current prototypes, the incoming photocurrent is directly mapped to the display using purely analog signal conditioning, which reduces the necessary circuitry. But it would be possible to integrate more advanced functionality within each pixel. For example, digital decoding circuitry in each pixel could be used to sense data modulated onto the incident light. This would provide a mechanism to change pixel sensitivity, erase pixel content, or control pixels with finer granularity. Our prototypes require a small amount of power in order to sense and render a new image, and this is supplied by a battery built into a "bulldog clip" that attaches to contacts on one edge of the display (see Figure 3b). However, we intend to provide power from behind the display in future iterations, allowing seamless tiling of displays. Ultimately, we'd like to leverage energy-harvesting techniques along the lines of the work done by Artem Dementyev and his colleagues.[7]

It's also important to note that autonomous pixel displays aren't limited to EPDs. We believe our architecture will generalize to a range of input and output modalities; for example, the addition of memory elements and

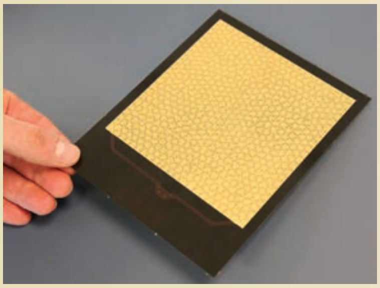
SPOTLIGHT

STEP-BY-STEP GUIDE: BUILDING A DISPLAY IN THE LAB

It can be remarkably straightforward to build a prototype display using regular lab equipment. In our case, the display backplane is a regular printed circuit board (PCB), as shown in Figure A1. The position, size, and shape of each pixel is defined by a series of electrodes running across the top surface of the PCB (see Figure A2). The electrophoretic display (EPD) media is supplied in sheet form and must be cut to match the overall size of the display (see Figure A3). We use ComFlec film from OED Technologies (www.oedtech.com).

Before applying the EPD film frontplane to the PCB backplane, it is necessary to prepare a connection to the indium tin oxide (ITO) layer on the front of the display. First the aluminum release sheet is removed from a corner (or edge) of the media to reveal the EPD material itself (see Figure B1). Then, the EPD material can be removed at that location; the exposed ITO is cleaned using isopropyl alcohol (see Figure B2). A small piece of double-sided self-adhesive tape is placed on the frontplane-driving segment (see Figure B3). We use electrically conductive tape type 9703 from 3M, but a variety of products are suitable, both anisotropic "Z" tape and regular electrically conductive adhesive tape. Alternatively, conductive epoxy such as CircuitWorks CW2400 can make a more robust connection, but it's harder to apply. When the conductive tape is in place, the remaining aluminum release sheet is removed (see Figure B4).

Finally, the frontplane and the PCB backplane are sandwiched together and bonded (see Figure C) by activating the heat-activated-film adhesive (which was applied to the media by the supplier). We do this using a regular T-shirt press at 110 °C. To make the display more robust and improve appearance, we apply a white vinyl bezel which we cut to size on a regular vinyl cutter. The finished product is shown in Figure 3 in the main text.



(1)

(2)

(3)

Figure A. Developing our prototype. (1) The backplane is a regular printed circuit board. (2) The position, size, and shape of each pixel is defined by a series of electrodes running across the top surface of the PCB. (3) The electrophoretic display (EPD) media must be cut to match the overall size of the display.

extra driving circuitry will allow for OLED-based[8] autonomous pixel displays, which would support full color, high resolution, and high framerates. At the same time, we envisage many pervasive display applications where a monochrome electrophoretic display is more than adequate. Such displays are unobtrusive and very low power yet capable of rendering useful digital content.

TOWARD DISPLAYS AS A MATERIAL

Our autonomous pixel prototypes introduce more flexibility in the design of displays by breaking away from the traditional grid-based display architecture. Our ultimate aim is to think of displays as a material rather than components. We'd like to empower designers of physical objects and spaces to craft and shape displays like they can with paper, textile, metals, and plastics. Display material could be tailored using well-established techniques like cutting, tiling, forming, and casting. This direction of research would complement similar efforts in giving sensors material-like properties.[9] The way forward involves implementing the circuitry present in our current prototypes using thin-film transistors on a truly flexible backplane. The resulting backplane would be laminated to the display media to produce a cuttable sheet of plastic containing millions of autonomous pixels.

This display material would be invaluable for creating objects with doubly curved, nondevelopable surfaces. Presently, these nondevelopable surfaces can only be approximated coarsely by tiling display panels; display bezels and visible gaps between the panels diminish the fidelity to which a particular surface can be realized. As illustrated in Figure 4, with display material, it would be possible to create display pieces with custom shapes that can much better approximate a curved surface when assembled. An even more radi-

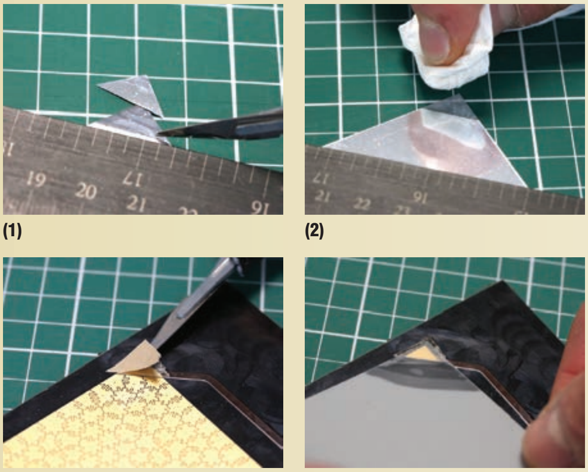


Figure B. Preparing a connection to the indium tin oxide (ITO) layer on the front of the display: (1) Remove the aluminum release sheet from a corner of the media to reveal the EPD material. (2) Remove the EPD material at that location and clean the exposed ITO using isopropyl alcohol. (3) Place a small piece of double-sided self-adhesive tape on the frontplane-driving segment. (4) Remove the remaining aluminum release sheet once the conductive tape is in place.

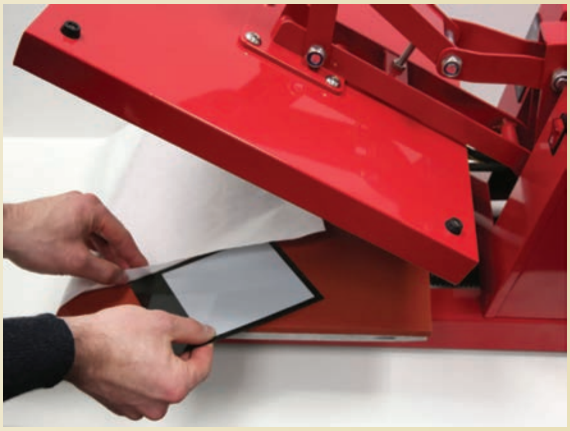


Figure C. The frontplane and the PCB backplane are sandwiched together and bonded by activating the heat-activated-film adhesive.

Driving the display is simply a matter of applying a voltage between any given pixel electrode and the ITO layer. The higher the voltage, the more quickly the white spheres move and the faster the display updates. We found that a 5V differential supports sub-1 second updates. The simplest approach is to drive each pixel directly, for example, using a microcontroller general-purpose input/output (GPIO) line. In this scheme, the ITO layer is first driven at 5 V so that any pixel electrodes that are then connected to 0 V will turn white due to the electric field created. Then, the ITO layer is connected to 0 V, and any pixels that need to become black are driven to 5 V. It's also possible to create more complex addressing schemes; for example, an active matrix display can be built by fitting discrete transistors behind each pixel electrode.

cal possibility is the ability to "spray" autonomous pixels onto arbitrarily shaped objects to make displays in unprecedented forms.

The autonomous pixel architecture described here and the concept of displays as a material are valuable steps in the evolution of digital displays. By lifting the restrictions associated with today's rectangular displays, the ergonomics of the devices and physical environments that incorporate displays could evolve in new ways. Ultimately, we hope that our work, combined with future developments in display

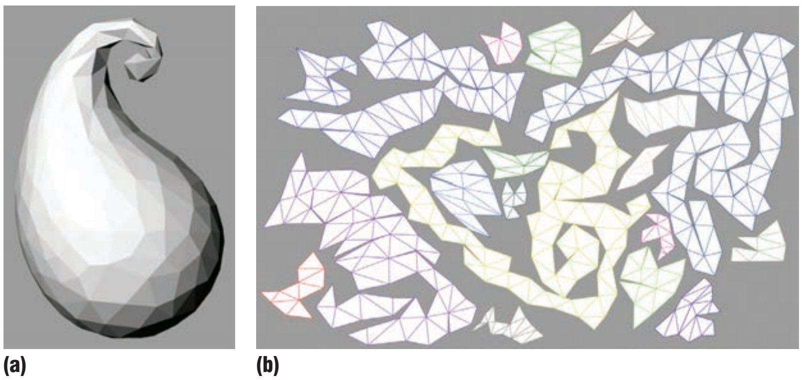


Figure 4. The benefits of display material: (a) Such material could support radical new form factors for displays, such as doubly curved surfaces. (b) You could also expand such a shape into a polygon mesh and cut the display material to the corresponding pattern before folding and forming the material back into the target shape.

SPOTLIGHT

technology, will lead to more informative, useful, and appealing devices and spaces. ◼

REFERENCES

1. K. O'Hara et al., *Situated Displays: Social and Interactional Aspects of Shared Display Technologies*, Springer, 2003.

2. S. Olberding, M. Wessely, and J. Steimle, "PrintScreen: Fabricating Highly Customizable Thin-Film Touch-Displays," *Proc. 27th Annual ACM Symp. User Interface Software and Technology* (UIST), 2014, pp. 281–290; http://dl.acm.org/citation.cfm?id=2647413.

3. M. Hoggenmueller and A. Wiethoff, "Blinking Lights and Other Revelations: Experiences Designing Hybrid Media Façades," *Proc. 4th Int'l Symp. Pervasive Displays* (PerDis), 2015, pp. 77–82; http://dl.acm.org/citation.cfm?id=2757725.

4. A. Chandler et al., "Toward Emergent Technology for Blended Public Displays," *Proc. 11th Int'l Conf. Ubiquitous Computing* (UbiComp), 2009, pp. 101–104; http://dl.acm.org/citation.cfm?id=1620562.

5. B. Comiskey et al., "An Electrophoretic Ink for All-Printed Reflective Electronic Displays," *Nature*, vol. 394, 1998, pp. 253–255.

6. M.E. Karagozler et al., "Paper Generators: Harvesting Energy from Touching, Rubbing and Sliding," *Proc. 26th Ann. ACM Symp. User Interface Software and Technology* (UIST), 2014, pp. 161–162; doi: 10.1145/2559206.2579480.

7. A. Dementyev et al., "Wirelessly Powered Bistable Display Tags," *Proc. 2013 ACM Int'l Joint Conf. Pervasive and Ubiquitous Computing* (UbiComp), 2013, pp. 383–386; http://dl.acm.org/citation.cfm?id=2493516.

8. A.B. Chwang et al., "Thin Film Encapsulated Flexible Organic Electroluminescent Displays," *Applied Physics Letters*, vol. 83, no. 3, 2003, pp. 413–415.

9. S. Olberding et al., "A Cuttable Multi-Touch Sensor," *Proc. 26th Ann. ACM Symp. User Interface Software and Technology* (UIST), 2013, pp. 245–254; http://dl.acm.org/citation.cfm?id=2502048.

David Sweeney is a creative design technologist at Microsoft Research. Contact him at dasween@microsoft.com.

Nicholas Chen is a researcher at Microsoft Research. Contact him at nchen@microsoft.com.

Steve Hodges leads the Sensors and Devices research group at Microsoft Research, Cambridge UK and is a visiting professor at the School of Computing Science, Newcastle University. Contact him at steve.hodges@microsoft.com.

Tobias Grosse-Puppendahl is a post-doc researcher at Microsoft Research. Contact him at tgp@microsoft.com.

This article originally appeared in IEEE Pervasive Computing, *vol.15, no. 3, 2016.*

EDITOR: Irena Bojanova, US National Institute of Standards and Technology, irena.bojanova@computer.org

The Internet of Things in Healthcare

Potential Applications and Challenges

Phillip A. Laplante, *Pennsylvania State University*
Nancy Laplante, *Widener University*

The Internet of Things (IoT) is a collective term for any one of the many networks of sensors, actuators, processors, and computers connected to the Internet. Healthcare applications for the IoT can potentially deliver comprehensive patient care in various settings, including acute (in-hospital), long-term (nursing homes), and community-based (typically, in-home).[1]

An IoT has the potential to accurately track people, equipment, specimens, supplies, or even service animals and analyze the data captured. With patients attached to sensors to measure vital signs and other biometric information, problems could be more rapidly diagnosed, a better quality of care given, and resources used more efficiently.

Applications for Healthcare in the IoT

The following represent just a handful of IoT healthcare application systems.

For the bulimia (eating disorder) patient in a hospital or home care setting, sensors in the patient's room could detect increased body temperature or blood pressure, or even the odor of vomit. Sensors could be used to detect exercise abuse such as excessive cardio training or accelerated walking activity as compared to walking at a normal pace. This data could provide valuable information in the diagnosis and management of the illness.

Consider a patient with Alzheimer's disease. Here, an IoT could employ geolocation to prevent wandering or other unwanted mobility behaviors. Often, patients with Alzheimer's suffer from comorbidities with other diseases, such as hypertension (high blood pressure), macular degeneration, or diabetes. Therefore, appropriate interconnected devices could capture data for monitoring the unique signs and symptoms of these conditions.

Safety and violence are real issues in healthcare today. There are numerous accounts of horizontal violence—for example, nurse against nurse—but also of violence from visitors or family toward healthcare providers or patients. Although healthcare institutions are equipped with video surveillance systems, an IoT could be another layer implementing a zero-tolerance policy. For example, tracking the movement of staff, patients, and visitors could provide warnings of aberrant or threatening behavior. Biometric sensors could be used to detect signs of aggression or stress in people who are entering or reside in these settings.

Monitoring in hospitals can take many paths. Staff might try to keep certain equipment, such as an IV pump or oxygen tanks, in their unit for future use. In a hospital, scarce shared equipment such as EKG machines, IV pumps, and patient-controlled analgesia (PCA) medication pumps could be tracked via an IoT. In addition, the use of such equipment would be of interest to individual units and administration—as well as insurance companies, including Medicare—in documenting the need for additional equipment. An IoT could also be used

 2469-7087/16/$33.00

to monitor equipment that needs to be refilled or calibrated, such as oxygen tanks, and to alert staff of such situations.

In an acute or long-term care setting, low-cost RFID or bar code tags allow many supplies to be tagged for scanning, making it easy to make charges to a patient's account. Such supplies can also be tracked using an IoT as they are either checked out from a repository or administered to a patient. In some cases, where an RFID tag is used, an item could be located more quickly, for example. Likely trackable items include one-time use supplies, such as dressings, catheters of different types, and personal care items. In a home setting, medical supplies could be marked with RFID tags to monitor use and alert the home care team when an item is being overused or supply is too low.

Researchers and practitioners envision many other IoT healthcare applications that could substantially improve patient care, optimize resource utilization, and save vast amounts of money—if only the systems could be built.

Challenges Ahead

The deployment of full-scale IoT systems for healthcare applications has not been reported in the literature. There are reports of experimental implementations—for instance, monitoring patients' biometric signs or identifying when a patient has fallen using accelerometer data.[2] To track workflow, Kyoto University Hospital implemented a real-time location system employing handheld barcode scanners, Bluetooth transmitters, a beacon relay system, and barcode tags on patients, nurses, and supplies.[3]

The absence of deployed systems in healthcare settings reflects both the novelty of the technology and the existence of significant problems. For example, there are technological problems, such as electromagnetic radiation effects and signal strength problems, inside hospitals. Changing the behavior of staff in acute and long-term care settings to cope with the new technology also presents some real challenges.

We stipulate that any healthcare system must be safe, and this quality must be incorporated into any system specifications in this domain. But one particular set of challenges to implementing real IoT healthcare systems must be addressed: security, privacy, and trust.

Security

IoT applications must be secure. Exposing any component of an IoT healthcare system to a hacker, whether a terrorist, disgruntled person, blackmailer, or any other malicious actor, can have deadly consequences. Many researchers are working on the problem of securing IoT systems completely, but because no system can be 100 percent secure, ethicists and medical, legal, security, and financial professionals must define and quantify acceptable risk.

Loss of Privacy

No class of exciting applications for the IoT epitomizes the tradeoffs between security and privacy and functionality and privacy more than those in healthcare. But privacy is of paramount importance because patients expect that certain private information will remain confidential. Therefore, IoT healthcare systems must allow for sharing information that is needed to provide high-quality care across the care continuum, while at the same time assuring privacy.

There are legal obligations to protect private information in a healthcare IoT. The US, for example, has the Health Insurance Portability and Accountability Act (HIPAA) of 1996, whereas systems in EU countries must comply with 1995's Data Protection Directive. There are other patient-specific privacy needs based on a wide range of factors, such as age, profession, religion, and personal preference. However, in today's high-tech healthcare environment, new concerns have been raised as to the relationship of HIPAA and the IoT that have yet to be resolved.

Trust

Information that is being delivered from sensors might appear to be correct, but could be corrupted somehow at the origin or during transmission, or deliberately altered by malware that can gain unwanted access to the IoT via the Internet. This corrupted information might then be used to make life and death decisions. How then, can we trust the information delivered to us in an IoT healthcare system? This problem has yet to be resolved.

Another form of trust relates to compassionate care. Caring is about a relationship, one that is forged between the patient, their family and community, and nurses and other healthcare professionals. Compassionate care for the sick is an expectation for all healthcare providers, but compassion is based on trust. For example, for the 14th straight year, nursing has been rated as the most honest, ethical profession.[4] This high rating has been built on a relationship that begins with trust and a personal connection with patients and the public. Nurses often struggle with balancing technology and patient contact, because technology can at times remove the nurse from the bedside. Conversely, technology has helped improve patient care by allowing for better assessment, surveillance,

This article originally appeared in IT Professional, *vol. 18, no. 3, 2016.*

and treatment. With the advent of the IoT in healthcare, nurses must incorporate technologies on many levels and determine the best use for their practice and how to use technology to achieve desired patient outcomes. Other healthcare professionals will have to do the same.

There are exciting applications of the IoT for healthcare that promise to enhance the patient experience, improve workflow, optimize the use of scarce resources, and provide substantial cost savings. But real, scalable systems have yet to be built, and significant obstacles must be overcome. These obstacles include technological issues, safety, and security, privacy, and trust.

The IoT is still a novel concept for most healthcare professionals, but its use in healthcare is inevitable. Although the IoT adds another layer to the debate of caring versus technology, we encourage deeper consideration of the benefits, and encourage nurses in particular to add their voices to the development and integration of technology. Nurses are at the patient's bedside and often are the ones who need to be most comfortable with these technologies. Patient tracking can occur through the IoT within hospitals and outside in our communities, where here again nurses will question whether technology is taking the "care" out of healthcare. Consider populations not easy to reach who are monitored via telehealth—although there can be less human interaction, nurses here are able to care for patients that otherwise would be forgotten. The IoT can allow for monitoring and communication that thus far has not been available.

In 2013, it was reported that there were two Internet-connected devices for each person, and predicted that by 2025, this number will exceed six.[5] As new IoT systems are developed and deployed, the challenge in healthcare is to improve patient care without a reduction in caring through reduced human contact with patients.

References

1. P.A. Laplante and N. Laplante, "A Structured Approach for Describing Healthcare Applications for The Internet of Things," *Proc. IEEE 2nd World Forum on Internet of Things*, to appear, 2016.
2. L. Catarinucci et al., "An IoT-Aware Architecture for Smart Healthcare Systems," *IEEE Internet of Things J.*, vol. 2, no. 6, 2015, pp. 515–526.
3. K. Sato et al., "Feasibility Assessment of Bluetooth-Based Location System for Workflow Analysis of Nursing Staff," *Trans. Japanese Society for Medical and Biological Eng.*, vol. 51, supplement, 2013, p. R-314.
4. "Nurses Rank as Most Honest, Ethical Profession for 14th Straight Year," press release, Am. Nurses Assoc. (ANA), 2015; www.prnewswire.com/news-releases/nurses-rank-as-most-honest-ethical-profession-for-14th-straight-year-300195781.html.
5. D. Skiba, "Emerging Technologies: The Internet of Things (IoT)," *Nursing Education Perspectives*, vol. 34, no. 1, 2013, pp. 63–64.

Phillip A. Laplante *is a professor of software engineering at the Pennsylvania State University. His research interests include the Internet of Things, software testing, requirements engineering, and software quality and management. Since 2010, Laplante has led the effort to develop a national licensing exam for software engineers. He received a PhD from Stevens Institute of Technology and an MBA from the University of Colorado, and is a Fellow of the IEEE and SPIE. Contact him at pal11@psu.edu.*

Nancy Laplante *is an associate professor of nursing at Widener University. Her research interests include healthcare applications for the Internet of Things, the image of nursing in media, and creating authentic presence in online nursing courses. Laplante received a PhD from Widener University. She is board certified in advanced holistic nursing. Contact her at nllaplante@mail.widener.edu.*

TORINO, ITALY

July 4-8, 2017

Hosted by the Politecnico di Torino, Turin, Italy

COMPSAC 2017 - ITiP Symposium

IT in Practice

www.compsac.org

ITiP is co-sponsored by: IT Professional

The COMPSAC 2017 symposium on IT in Practice (ITiP), co-sponsored by *IT Professional*, serves the interests and needs of IT practitioners, application developers, IT managers, and CXOs from business, industry, and government, as well as academics and researchers focusing on application-oriented work. The symposium focuses on diverse elements necessary to successfully architect, design, engineer, configure, and field a range of IT applications in several domains. It is an interactive forum for sharing and gaining valuable insights and experiences and for active engagement among technical professionals around the globe.

We solicit submissions on a range of topics related to the symposium theme, including:

- Information assurance and security
- IT systems model and infrastructure
- Human computer interaction
- Networking platform technologies
- Web systems and technologies, adaptability, professionalism
- Data, information, and knowledge management
- Collaborative computing
- Enterprise systems and architecture
- IT project management
- IT audit and controls
- IT application design & validation
- Securing enterprise information system
- Enterprise data analytics
- Data mining and business intelligence
- Social computing
- Big data, IoT, and mobile crowdsourcing in practice
- IT practice in business, healthcare, industry, and other sectors
- Design, deployment and management of large scale IT systems
- Real world issues, concerns, challenges
- Innovative IT applications
- Smart cities and Cyber-Physcial Systems
- Emerging trends in enterprise in IT

Paper guidelines can be found at ***www.compsac.org***. In addition to regular papers we welcome, particularly from industry authors, short papers (6 pages) and fast abstracts (2 pages). Authors are invited to submit original, unpublished research work, as well as industrial practice reports. Simultaneous submission to other publication venues is not permitted. In accordance with IEEE policy, submitted manuscripts will be checked for plagiarism; instances of alleged misconduct will be handled according to the IEEE Publication Services and Product Board Operations Manual. Detailed instructions for electronic paper submission, panel, workshop, and tutorial proposals, fast abstracts, industry papers, poster papers, doctoral symposium, and the review process will be available at www.compsac.org.

IMPORTANT DATES

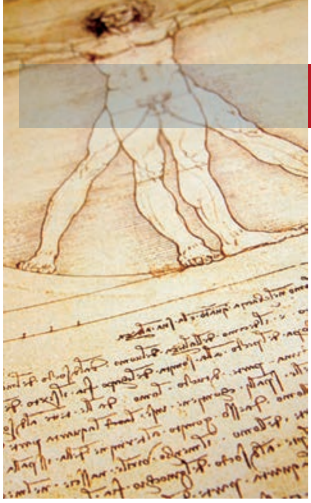
Main Conference Papers Due:
13 January 2017
Author Notification:
27 March 2017
Registration Due:
9 May 2017

ITiP SYMPOSIUM ORGANIZERS

Program Co-Chairs:
Maria R. Lee, Shih Chien University, Taiwan
Bin Guo, Northwestern Polytechnical University, China
Steering Chair:
San Murugesan, Editor in Chief, *IT Professional*; BRITE Professional Services, Australia

COMPSAC 2017 ORGANIZERS

Advisory Committee Chair:
Sorel Reisman, California State University, USA
Steering Committee Chair:
Sheikh Iqbal Ahamed, Marquette University, USA
General Chairs:
Claudio Demartini, Politecnico di Torino, Turin, Italy
Tom Conte, Georgia Institute of Technology, USA
Program Chairs in Chief:
Lin Liu, Tsinghua University, China
William Claycomb, Carnegie Mellon University, USA
Motonori Nakamura, NII, Japan
Edmundo Tovar, Universidad Politecnica de Madrid, Spain



Editors: **Robert R. Hoffman, Ken Ford, and Matthew Johnson,** Florida Institute for Human and Machine Cognition, rhoffman@ihmc.us

Expertise: A Second Look

David Z. Hambrick, *Michigan State University*
Robert R. Hoffman, *Institute for Human and Machine Cognition*

This essay hearkens to the origins of *IEEE Intelligent Systems* in the era of "expert systems," which witnessed the emergence of notions of knowledge capture, representation, and management. In this essay, we summarize recent work that has called into question earlier definitions of expertise, both conceptual (what are the capacities of experts?) and operational (precisely how do we identify experts and measure expertise?).

Practice with Zeal

It is an undeniable and readily observable fact that some people are much better than others at certain complex cognitive tasks. Marion Tinsley won the World Checkers Championship each of the 20 times he entered it and lost only five games to a human opponent in his 45-year career.[1] Dan Feyer won the American Crossword Puzzle Tournament six years in a row and can solve a Saturday *New York Times* crossword puzzle (regarded as a "form of mental cruelty") in about five minutes.[2] It is more difficult to quantify skill in professions such as law, science, economics, forecasting, and medicine, but individual differences in expertise in these and many other domains are no less readily apparent.[3]

In 1869, Francis Galton argued that genius in science, music, art, and other professions arises almost inevitably from natural ability.[4] In 1912, educational psychologist Edward Thorndike countered that "we stay far below our own possibilities in almost everything that we do … not because proper practice would not improve us further, but because we do not take the training or because we take it with too little zeal."[5] Thus, the pendulum swung between the view that experts are "born" and the view that they are "made."

The experts-are-made view has dominated cognitively oriented research on expertise for the past half-century. In his pioneering research, Adriaan de Groot had chess players verbalize their thoughts when given "choice-of-move" problems.[6] Contrary to the popular view that chess skill reflects the ability to think ahead in a chess game, de Groot found that grandmasters were no different than less-skilled players on a measure of move depth. Instead, de Groot argued that the chess master "immediately 'sees' the core of the problem in the position, whereas the expert player finds it with difficulty—or misses it completely."

Building on that work, William Chase and Herbert Simon found that strong chess players recalled more of the permissible game positions than weaker players did, but this advantage did not extend to random positions.[7] They concluded that the primary factor underlying chess skill is not superior short-term memory, but a "vocabulary" of games positions that automatically elicit candidate moves. More generally, they noted that although there must be a talent for chess, "the overriding factor in chess skill is practice." Additional research reinforced this view. Through more than 200 hours of training, an undergraduate student was able to increase his recall of random digits by a factor of 10, from a typical 7 to 79 digits. Verbal reports revealed that the student had learned to encode groups of digits as running times.[8] Another study documented the unique strategies of a waiter who could remember upwards of 20 dinner orders without taking notes.[9]

It takes a lot of experience not just to develop skills but to acquire a rich knowledge of domain concepts and principles. Edward Feigenbaum captured this view in what he called the Knowledge Principle:[10]

> [A] system exhibits intelligent understanding and action at a high level of competence primarily because of the specific knowledge that it can bring to bear: the concepts, representations, facts, heuristics, models, and methods of its domain of endeavor. A corollary of the [Knowledge Principle] is that the reasoning processes of intelligent systems are generally weak and not the primary source of power.

This tradition of research affirmed a rule of thumb that was first proposed by John Hayes in 1985: that the achievement of expertise takes 10 years (or about 10,000 hours) of practice.[11] Popularizations have transformed the rule into a cultural myth.[12,13]

The Deliberate Practice View

K. Anders Ericsson and colleagues proposed that individual differences in performance in domains such as music and chess largely reflect differences in the amount of time people have spent in "deliberate practice."[14] Echoing Thorndike's notion of practice with zeal, deliberate practice was defined as motivated engagement in training activities that are specifically designed to improve performance in a domain, and which are difficult and therefore not necessarily pleasurable. To test their theory, violinists from a Berlin music academy estimated the amount of hours per week they had devoted to deliberate practice each year since having taken up violin. The students whom the faculty had nominated as the "best" violinists had accumulated an average of more than 10,000 hours of deliberate practice by the age of 20, which was about 2,500 more hours than the average for the "good" violinists and about 5,000 more hours than the average for the least-accomplished violinists in a group of music education students. "Expert" pianists had similarly accumulated an average of more than 10,000 hours of deliberate practice by the age of 20, compared to about 2,000 hours for "amateur" pianists.

The positive impact of the deliberate practice view has been to emphasize the importance of operationally defining expertise. All too often, studies on "expert–novice differences" involve individuals who are experienced in a domain, usually just measured in years. Worse, the vast bulk of the literature on expertise assumes that the mass of humanity can be neatly divided into experts versus novices. There are many levels, and sublevels, each of which must be nailed down in any empirically robust proficiency scaling process.[15]

Deliberate practice is arguably necessary for the achievement of expertise. But there is more to the deliberate practice viewpoint than this: the claim is that the achievement of expertise is not a function of innate ability. Ericsson and his colleagues concluded the following (emphasis added):[14]

> [E]xpert performance is qualitatively different from normal performance … . [E]xpert performers have characteristics and abilities that are qualitatively different from or at least outside the range of those of normal adults. *However, we deny that these differences are immutable, that is, due to innate talent.* Only a few exceptions, most notably height, are genetically prescribed. Instead, we argue that the differences between expert performers and normal adults reflect a life-long period of deliberate effort to improve performance in a specific domain.

This stance triggered considerable debate.[16–19] Philip Ackerman threw down the gauntlet,[20] saying

> until Ericsson shows cognitive expertise development in a randomly selected group of subjects, including those with moderate mental retardation, there is no reason to believe that such development can be accomplished.

The first thing to determine is exactly how much of the variance in proficiency can be accounted for by deliberate practice. One of us (ZH) conducted a reanalysis of the relationship between deliberate practice and performance in two of the most popular domains for expertise research—music and chess.[21,22] To be included in the reanalysis, a study must have used a continuous measure of some activity interpretable as deliberate practice and some sort of measure of performance, and a correlation between the measures had to be reported. Six studies of chess and eight studies of music met the criteria. In the chess studies, the measure of deliberate practice was time at chess study, and the measure of performance was chess rating (that is, the Elo rating). In the music studies, the measure of deliberate practice was time at practice, and the measure of performance varied by study (for example, audition ratings or sight-reading performance).

Not surprisingly, high levels of deliberate practice were associated with superior performance. However, even after correcting for the unreliability of the measures of deliberate practice and performance (average corrected $r = 0.54$), deliberate practice left more of the variance in performance unexplained than it explained. The average amount of variance in performance accounted for was 34 percent for the chess studies and 30 percent for the music studies.

None of the studies considered "the effects of forgetting, injuries, and accidents, along with the differential effects of different types of practice at different ages and levels of expert performance,"[23] including the studies that had previously been cited as support for the importance of deliberate practice.

A subsequent meta-analysis looked at all the major domains in which the relationship between deliberate practice and performance had been studied.[24] To be included, a study had to collect a measure reflecting one or more activities interpretable as deliberate practice and a measure of performance, and report an effect size or provide enough information for effect size to be computed. Through a search of more than 9,000 published and unpublished documents, 88 studies met these criteria, with a total of 157 effect sizes and a total sample size of more than 11,000. Nearly all of these effect sizes were positive, indicating again that high levels of deliberate practice are associated with superior performance. Again, however, deliberate practice left more of the variance in performance unexplained than explained—26 percent in studies of games and 21 percent in studies of music. Ericsson argued that only one of the studies included in the meta-analysis met his criteria for assessing the relationship of deliberate practice and performance[25]—his earlier study of pianists.[14] But the meta-analysis included studies that Ericsson and others had previously cited as support for the deliberate practice view.

The conclusion is that deliberate practice is an important and likely necessary factor in the achievement of expertise, but other factors are also important.

What Else Is Going On?

Individuals differ in the amount of deliberate practice that they need in order to reach a given proficiency level. Results of a study on chess by Fernand Gobet and Guillermo Campitelli showed that there was considerable variability in the amount of deliberate practice to first reach "master" status—from 3,016 hours to 23,608 hours.[26] Some players had not reached master status, despite having accumulated more than 10,000 hours of deliberate practice. Of the 16 masters in the sample, nearly a third (31.3 percent) had accumulated less deliberate practice than the mean amount for the expert group, whose members were one skill level down. In the other direction, of the 31 intermediates, a quarter (25.8 percent) had accumulated more deliberate practice than the mean amount for the experts, who were one skill level up.

Just saying that there are "individual differences" is to merely acknowledge that there is a mystery. The age at which an individual starts in a domain may be one explanatory factor. There is evidence that starting at a young age in a domain has benefits for later performance, even after accounting for deliberate practice.[26] There is also growing evidence that basic cognitive abilities contribute to individual differences in expertise, at least under some circumstances.

One ability construct of considerable interest to expertise researchers is working memory capacity. Working memory capacity can be thought of as the ability to temporarily maintain information in a highly active and accessible state, and it is typically measured with "complex span" tasks in which the goal is to carry out some mental operation while remembering a set of items.[27] For example, in operation span, the participant solves arithmetic equations while remembering a string of letters. As measured with tasks such as these, working memory capacity correlates moderately and positively with individual differences in reading comprehension, problem solving, multitasking, and other complex tasks.

The deliberate practice view is that working memory capacity and other basic abilities are important only in the initial stages of skill acquisition, and that the effects of these factors on performance can be "circumvented" through skills and strategies that develop with deliberate practice. "[T]he effects of extended deliberate practice are more far-reaching than is commonly believed. Performers can acquire skills that circumvent basic limits on working memory capacity and sequential processing."[28]

One of us (ZH) tested this circumvention-of-limits hypothesis in a study of pianists.[29] The participants, who ranged in skill from beginner to professional, completed a questionnaire to assess their accumulated amount of deliberate practice, and they took standard tests of working memory capacity. They then completed a sight-reading task in which they attempted to play pieces of music without preparation. Deliberate practice accounted for 45 percent of the variance in performance. Working memory capacity accounted for an additional 7.4 percent of the variance—a much smaller but still statistically and practically significant effect. Moreover, inconsistent with the circumvention-of-limits hypothesis, working memory capacity was no less important a predictor of performance for pianists having had thousands of hours of deliberate practice than it was for beginners.

Other cognitive abilities have also been found to correlate with performance in specific domains. Results of a study of the Chinese board game Go are of particular note given the large sample size and wide range of skill in this study.[30] On standard cognitive ability tests, participants representing a wide range of skill had scores that correlated near zero with tournament Go rating, but they correlated significantly and positively with scores on Go-related tasks. There was a strong positive relationship between Go skill and what

The expert's judgments are accurate, reliable, and useful:

- The expert's performance shows consummate skill (that is, more effective and/or qualitatively different strategies) and economy of effort (that is, more efficient).
- The expert possesses knowledge that is fine grained, detailed, and highly organized.
- The expert uses a large repertoire of strategies for reasoning about tasks and conducting tasks.
- For routine activities and familiar cases, experts display signs of "automaticity" and "recognition-primed decision making," wherein the expert seems to be carrying out a task without significant cognitive load, and conscious processing is reserved for strategic control, rare or unusual cases, and/or more complex activities.
- The expert has refined pattern perception skills and can apprehend meaningful cues, relationships, and patterns that nonexperts cannot.
- Experts can recognize aspects of a problem that make it novel or unusual, and will bring special strategies to bear to solve "tough cases."
- The expert is highly regarded by peers.
- The expert knows that his or her knowledge is constantly changing and continually contingent.
- The expert forms rich mental models of cases or situations to support sense-making and anticipatory thinking.
- The expert can create procedures and conceptual distinctions, sometimes on the fly as new challenges appear.
- The expert can effectively manage resources under conditions of high stakes, high risk, and high stress.
- Typically, experts have special knowledge or abilities derived from extensive experience with subdomains.
- Typically, individuals who achieve expertise have benefitted from tutelage under an expert mentor.

Figure 1. A conceptual definition of expertise.

psychologists call "fluid intelligence." Moreover, the relationship was similar in magnitude across beginner, intermediate, and expert skill levels.

Child prodigies who perform at a level well beyond that of their peers have been shown to score at or above the 99th percentile for working memory.[31] A study involving more than 10,000 twins found no evidence for a causal influence of the amount of time devoted to music practice on music aptitude. Although some music skills can only be acquired through practice (for example, reading music or fingering a keyboard), this finding suggests that some abilities underlying musical expertise are not changed through practice.

Taken together, these findings suggest that basic abilities can be an important factor in the achievement of expertise. This is not to say, however, that there are no circumstances under which the influence of basic abilities are attenuated at high levels of skill, as Ericsson and his colleagues hypothesized. A study of geologists found that visuospatial ability predicted success in a highly realistic task in which the goal was to map an area's geological structure, but only for individuals having lower proficiency, and not for experts.[32] The challenge for future research is to identify task and situational factors that moderate the interplay between acquired characteristics and basic abilities.

Implications for Intelligent Systems

What are the implications of these ideas and findings for the design of intelligent systems? One implication we must jettison, once and for all, is the assumption that the world can be neatly divided into experts versus novices. Taking this further, we should not be so wedded to the word "expert." We have taken this word, conveniently delivered to us by our language, and reified it. Proficiency scaling—whether to identify experts for knowledge capture or to design training systems—must use rigorous empirical methods and objective performance measures to define domain-appropriate proficiency scales spanning novice, apprentice, journeyman, and expert levels, and in some cases sublevels.[15]

Another implication is to keep a focus on the conceptual, as well as the operational definition of expertise. A best conceptual definition must capture the full richness of expertise, including cognitive aspects and social aspects, as shown in Figure 1.[15,33]

We agree with the idea that one should not claim anything about an individual's expertise, or expertise in general, without having first adduced convincing evidence that the individuals under study qualify as showing high or highest levels of proficiency, as Ericsson has rightly insisted. But if expertise is defined solely in terms of "superior performance on representative tasks conducted under controlled circumstances," then huge swaths of human activity are potentially removed from empirical scrutiny. Some domains have few practitioners, and you have to "go with what you can get" when doing knowledge capture or conducting research on proficiency levels. In some domains, it is hard to come up with a single best measure of performance, and it would be misleading to rely on any single performance measure anyway. For example, there are relatively few world-class art-appraisal experts. They sometimes disagree in their judgments—for example, about whether a particular painting is a long-lost masterpiece by some famous artist. But this does not negate their claims to expertise, since they show other features we attribute to experts, such as the ability to determine whether individual brush strokes represent a particular artist's style.[34]

In some domains, tasks are constantly morphing and new challenges emerging. There are even entirely new, emergent domains. Perhaps not surprisingly, these are often domains of great societal and national importance, such as cyberdefense. Thus, proficiency scaling can and should always rely on a multimethod approach, involving sociometric analysis and analyses of career experiences as well as performance measures.[15]

Looking beyond chess and violin—however important and valuable to society those might be—we need methods for studying high proficiency in dynamic and messy domains. We need not engage in the methodolatrist's retreat to study only that which can be readily brought into the academic laboratory. We need to bring the world into the lab, surely. But we also need to bring the lab into the world.

Experts stand out because of their superior performance and unique capabilities (the first two bullets in Figure 1). They have therefore served as the benchmark for "intelligence" in the development of intelligent systems (the second two bullets in Figure 1). All of the other bullets in Figure 1 are potential leverage points for the application of intelligent systems and for the extension of the intelligence of intelligent systems. Especially clear as a leverage point is training, as indicated in the final bullet in Figure 1. The concept of "accelerated expertise" is premised on the idea that different abilities and capacities come into play across the levels and sublevels of proficiency, meaning that training must be level-appropriate.[15] Especially promising is the idea of going beyond intelligent tutoring to intelligent mentoring.[35] ■

References

1. G. Belsky, "A Checkered Career," *Sports Illustrated*, 28 Dec. 1992, pp. 140–148.
2. D. Grady, "Across and Down, the Wizard Who Is Fastest of All," *New York Times*, 6 Dec. 2010.
3. K.A. Ericsson et al., eds., *The Cambridge Handbook of Expertise and Expert Performance*, Cambridge Univ. Press, 2006.
4. F. Galton, *Hereditary Genius*, Macmillan, 1869.
5. E. Thorndike, *Education: A First Book*, Macmillan, 1912, p. 108.
6. A. de Groot, *Thought and Choice in Chess*, Mouton Publishers, 1946.
7. W.G. Chase and H.A. Simon, "Perception in Chess," *Cognitive Psychology*, vol. 4, no. 1, 1973, pp. 55–81.
8. K.A. Ericsson, W.G. Chase, and S. Faloon, "Acquisition of a Memory Skill," *Science*, vol. 208, no. 4448, 1980, pp. 1181–1182.
9. K.A. Ericsson and P.G. Polson, "An Experimental Analysis of the Mechanisms of a Memory Skill," *J. Experimental Psychology: Learning, Memory, and Cognition*, vol. 14, no. 2, 1988, pp. 305–316.
10. E.A. Feigenbaum, "What Hath Simon Wrought?" *Complex Information Processing: The Impact of Herbert A. Simon*, D. Klahr and K. Kotovsky, eds., Erlbaum, 1989, pp. 165–180.
11. J.R. Hayes, "Three Problems in Teaching General Skills," *Thinking and Learning Skills, Vol. 2: Research and Open Questions*, S.F. Chipman, J.W. Segal, and R. Glaser, eds., Erlbaum, 1985, pp. 391–405.
12. G. Colvin, *Talent Is Overrated: What Really Separates World-Class Performers from Everybody Else*, Penguin, 2010.
13. M. Gladwell, *Outliers: The Story of Success*, Little, Brown, and Co., 2008.
14. K.A. Ericsson, R.T. Krampe, and C. Tesch-Römer, "The Role of Deliberate Practice in the Acquisition of Expert Performance," *Psychological Rev.*, vol. 100, no. 3, 1993, pp. 363–406.
15. R.R. Hoffman et al., *Accelerated Expertise: Training for High Proficiency in a Complex World*, CRC Press, 2014.
16. H. Gardner, "'Expert Performance: Its Structure and Acquisition': Comment," *Am. Psychologist*, vol. 50, no. 9, 1995, pp. 802–803.
17. W. Schneider, "Innate Talent or Deliberate Practice as Determinants of Exceptional Performance: Are We Asking the Right Question?" *Behavioral and Brain Sciences*, vol. 21, 1998, pp. 423–424.
18. D.K. Simonton, "Talent and Its Development: An Emergenic and Epigenetic Model," *Psychological Rev.*, vol. 106, no. 3, 1999, pp. 435–457.
19. R.J. Sternberg, "Costs of Expertise," *The Road to Excellence: The Acquisition of Expert Performance in the Arts and Sciences, Sports, and Games*, Erlbaum, K.A. Ericsson, ed., 1996, pp. 347–354.
20. P.L. Ackerman, "Facts Are Stubborn Things," *Intelligence*, vol. 45, July/Aug. 2014, pp. 104–106.
21. D.Z. Hambrick et al., "Deliberate Practice: Is That All it Takes to Become an Expert?" *Intelligence*, vol. 45, no. 1, 2014, pp. 34–45.
22. F. Platz et al., "The Influence of Deliberate Practice on Musical Achievement: A Meta-Analysis," *Frontiers in Psychology*, 2014; doi:10.3389/fpsyg.2014.00646.
23. K.A. Ericsson, "Why Expert Performance Is Special and Cannot Be Extrapolated from Studies of Performance in the General Population: A Response to Criticisms," *Intelligence*, vol. 45, July/Aug. 2014, pp. 81–103.
24. B.N. Macnamara, D.Z. Hambrick, and F.L. Oswald, "Deliberate Practice and Performance in Music, Games, Sports, Education, and Professions: A Meta-Analysis," *Psychological Science*, vol. 25, 2014, pp. 1608–1618.
25. K.A. Ericsson, "Challenges for the Estimation of an Upper-Bound on Relations

This article originally appeared in IEEE Intelligent Systems, *vol. 31, no. 4, 2016*

Between Accumulated Deliberate Practice and the Associated Performance of Novices and Experts: Comments on Macnemara (sic), Hambrick, and Oswald's (2014) Published Meta-Analysis," unpublished manuscript, 2014; http://psy.fsu.edu/faculty/ericsson/ericsson.hp.html.
26. F. Gobet and G. Campitelli, "The Role of Domain-Specific Practice, Handedness, and Starting Age in Chess," *Developmental Psychology*, vol. 43, no. 1, 2007, pp. 159–172.
27. R.W. Engle, "WMC as Executive Attention," *Current Directions in Psychological Science*, vol. 11, no. 1, 2002, pp. 19–23.
28. K.A. Ericsson and N. Charness, "Expert Performance: Its Structure and Acquisition," *Am. Psychology*, vol. 49, 1994, pp. 725–747.
29. E.J. Meinz and D.Z. Hambrick, "Deliberate Practice is Necessary but Not Sufficient to Explain Individual Differences in Piano Sight-Reading Skill: The Role of Working Memory Capacity," *Psychological Science*, vol. 21, 2010, pp. 914–919.
30. H. Masunaga and J. Horn, "Expertise and Age-Related Changes in Components of Intelligence," *Psychology and Aging*, vol. 16, no. 2, 2001, pp. 293–311.
31. J. Ruthsatz and J.B. Urbach, "Child Prodigy: A Novel Cognitive Profile Places Elevated General Intelligence, Exceptional Working Memory and Attention to Detail at the Root of Prodigiousness," *Intelligence*, vol. 40, 2012, pp. 419–426.
32. D.Z. Hambrick et al., "A Test of the Circumvention-of-Limits Hypothesis in Scientific Problem Solving: The Case of Geological Bedrock Mapping," *J. Experimental Psychology: General*, vol. 141, 2012, pp. 397–403.
33. R.R. Hoffman, "How Can Expertise Be Defined? Implications of Research from Cognitive Psychology," *Exploring Expertise*, R. Williams, W. Faulkner, and J. Fleck, eds., Macmillan, 1998, pp. 81–100.
34. L. Manly, "5,200 or $5.2 Million? It's All in How It's Framed," *New York Times*, 7 Mar. 2015.
35. R.R. Hoffman and P. Ward, "Mentoring: A Leverage Point for Intelligent Systems?" *IEEE Intelligent Systems*, Sept./Oct. 2015, pp. 78–84.

David Z. Hambrick is a professor in the Department of Psychology at Michigan State University. Contact him at scienceofexpertise.com

Robert R. Hoffman is a senior research scientist at the Institute for Human and Machine Cognition. Contact him at rhoffman@ihmc.us.v

Meaning and Persuasion: The Personal Computer and Economic Education

Caroline Jack
Cornell University

Editor: Bradley Fidler

In the last days of 1982, the corporate-funded business education nonprofit Junior Achievement (JA) began distributing 121 donated personal computers—Xerox 820s, Hewlett Packard HP-86s, and IBM personal computers—to classrooms in 25 cities across the United States.[1] The donated machines were part of JA's new high school course, Applied Economics. The course curriculum had been designed with the goal of teaching economic principles, business skills, and the appreciation of private enterprise to high school students.[2] The personal computer, JA executives hoped, would draw student interest toward the course and its perspective on American economic life. The course, including its classroom lectures, in-class activities, computerized bookkeeping, and management simulation software, would present a vision of American economic life in which market forces organized the economy, with necessary but decidedly minimal interventions by organized labor or the state.

The research for my dissertation focused on the production of these kinds of corporate-sponsored "economic education" media: television documentaries, public service announcements, and school curricula that sponsors and producers hoped would instill an appreciation of private enterprise in the American public.[3] Sponsored economic education media productions were particularly prevalent from the New Deal era through the late Cold War period.[4] Sometimes, corporations directly sponsored and produced economic education materials. More often, however, nonprofit advocacy, outreach, or education groups acted as institutional intermediaries, taking corporate grants or sponsorships to fund media production. These media productions—ranging across diverse forms such as pamphlets, films, television programs, filmstrips, textbooks, and digital software—were efforts to maintain and buttress the social legitimacy of capitalism and private enterprise in the public imagination.

Many of the groups involved in economic education were interested in what personal computers could help them communicate to the public. The act of embracing computer technology could help an economic education group signal a forward-looking orientation. Furthermore, economic education groups were interested in software because it could represent complex ideas: from their perspective, software was communicative and therefore potentially persuasive.

Maintaining Meaning

Recent scholarship has emphasized the importance of the *maintainers*—that is, the people who "keep ordinary existence going" by repairing and maintaining our technological systems.[5] Technological systems, however, are not the only systems that require maintenance. Systems of cultural meaning—that is to say, ideologies—must also be carefully maintained to accommodate changing conditions and avoid ideological breakdowns that could shift the distribution of power, wealth, and perceived social legitimacy within a society. In other words, the advocacy groups, education nonprofits, and corporate sponsors involved in making economic education media were maintainers of a different sort. They used media in their efforts to repair and maintain ideologies, in the process framing a particular set of American capitalist institutional practices and social values as "ordinary existence."

We can examine these maintenance efforts by exploring not only the traditional media such organizations produced, but also their turn to and deployment of computers. JA's adoption of computing in the Applied Economics curriculum illustrates how this nonprofit business education group conceptualized personal computer hardware and software: as a means of drawing the attention of disengaged students toward an appreciation of free market perspectives, and as a means of aligning a learning-by-doing tradition with the rhythms and norms of the secondary school classroom.

This is not to suggest that there is something intrinsic or natural that links computing to capitalism in the abstract or to the many and varied ways Americans practice capitalism. Rather, the case provides one more example of the myriad ways computers were drawn into, and became modes of expression for, political and economic ideologies. Links drawn between capitalist ideology and computing include Thomas Streeter's critique of Silicon Valley's "two guys in a garage" mythos, for example, which draws out the romantic individualism embedded in 1980s personal computing lore. Similarly, Fred Turner's unraveling of the 1990s cyber-elite's countercultural roots helps illuminate the emergence of techno-libertarianism.[6] Other works such as Eden Medina's account of socialist computing visions in Allende's Chile and Benjamin Peters' chronicling of Soviet attempts to build a nationwide networked system of computers, however, show that noncapitalist

 Published by the IEEE Computer Society 2469-7087/16/$33.00

political and social meanings, too, have been attached to computers.[7,8] In each case, digital technologies became entwined with the work of maintaining, updating, and enacting existing ideological configurations.

Digital technologies became entwined with the work of maintaining, updating, and enacting existing ideological configurations.

The Company Experience

For the 62 years prior to the debut of Applied Economics, JA had offered after-school programs to high school age participants as a way to foster good relations between business and the community. JA's signature offering was the "company experience," in which teenaged participants ran a small door-to-door sales business with the sponsorship and guidance of a local businessperson. Through these activities, teenagers could learn manufacturing, marketing, and managerial skills that, the organization promised, would lead to good jobs after graduation. However, participation in JA's afterschool programs dropped in the mid-1970s. A confluence of economic and social factors left JA facing lowered youth enrollments and high operational costs.[9]

JA experimented with a variety of diversifying strategies that eventually led to its adoption of computers for classroom use. Most notably, it developed Project Business, an in-school program for junior high school students that took place on school grounds, thus cutting facility costs and reaching a wider audience. Project Business's success inspired JA executives to develop Applied Economics, a semester-long high school course designed to capture low-engagement students with an experience-rich curriculum. Alongside traditional lectures on the principles of managerial economics, students assembled small gift products (such as novelty memo pads) and sold them to classmates over the course of the semester. Students used the donated computers to keep their company books and to play competitive business simulation games using custom software inspired by the simulation games then in use at Harvard Business School.[9]

These activities set Applied Economics apart from other high school level economics curricula, drawing both praise and consternation. A majority of students in the pilot study reported that the course was enjoyable, useful, and interesting.[10] However, educators involved in the pilot and subsequent rollout of the program reported that recreating the company experience in the classroom was unwieldy. Too much time went to assembling merchandise, teachers reported, and some administrators banned the company experience activity altogether due to concerns that students' selling activities conflicted with preexisting in-school selling projects.[11]

The personal computer, already present and in use for bookkeeping and simulation games, offered a solution. In 1986, JA developed Computerized Student Company Software for use in Applied Economics that allowed students to make hypothetical sourcing, marketing, and management decisions without the need to handle cash or inventory. Instead, students used the software to plan budgets, place orders for raw materials, and navigate virtual representations of assembly lines and warehouses.[12] In essence, students shifted their use of classroom time from assembling goods for sale to making decisions in concert with computer software. The software gave students opportunities to practice technological and decision-making skills that would help them prepare for the demands of managerial jobs, but it reduced their firsthand experience with job roles such as product assembly and sales. In this adjustment, white-collar activities came to the forefront while blue-collar activities—specifically, the embodied activities of laboring, the interpersonal activities of selling, and the organizational activities of negotiating with management as a salesperson or laborer (either individually or collectively)—receded into the simulation software. In the years that followed, JA continued to experiment, in time introducing new live-action activities in which students could choose from a wide range of blue-collar and white-collar roles. Even so, using software to bring the company experience into alignment with existing classroom rhythms had representational consequences: students using the Computerized Student Company Software experienced the workplace mainly from a managerial perspective.

The details of media production, distribution, and use are often left out of these historical accounts.

Mapping Intermediaries' Computer Histories

In a recent *Annals* Think Piece article, Joy Rankin called for a history of social computing, one that considers the social and cultural aspects of computing.[13] I take her call as a prompt to consider how organizations have adopted computer hardware and software as elements in wider campaigns to communicate with the public. For example, business historians, labor historians, and historians of capitalism are creating a growing body of literature on American industrial advocacy in the 20th century; many mention the sponsored economic education media productions that reflected, and aimed to maintain, favorable public understandings of private enterprise. The details of media production, distribution, and use, however, are often left out of these historical accounts. Historians of computing can enrich this discourse by exploring how corporate-sponsored nonprofit groups made use of software in their public outreach efforts. More generally, we can map how computing was enfolded into ongoing ideological maintenance efforts, and we can attend to the ways institutional intermediaries perceived personal computer hardware and software as persuasive technologies.

This type of scholarly inquiry can bring to light how efforts to maintain systems of social and cultural meaning involved technology in complex and contingent ways. JA's initial decision to include donated computers and custom software in the Applied Economics curriculum was premised on the notion that students would be more interested in an economics course that offered engaging, technologically mediated experiences of business and commerce. Although one of JA's goals for Applied Economics was to foster an appreciation of private enterprise, the inclusion of blue-collar and white-collar job roles in JA's subsequent offerings suggest that Applied Economics was not necessarily a strategic, purposeful effort to make of every student a corporate manager in training. Yet, the need to respond to the temporal and spatial constraints of the classroom prompted the development of the Computerized Student Company Software, which relegated the activities of laborers and salespeople to the background. In this sense, the adoption of computers and custom software in Applied Economics had a variety of ideological implications, only some of which may have been intended. Computer historians are well placed to excavate the factors that may have contributed to the Computerized Student Company Software (and other applications like it) in its final form, including institutional intermediaries' organizational cultures, their perceptions of what software could or should do, and the professional and organizational backgrounds that may have shaped those perceptions.

As the personal computer became associated with business identities in the final decades of the 20th century, software became an increasingly viable site for reflecting, maintaining, and shaping cultural understandings of business and economics. Mapping the history of how business advocacy and education groups came to adopt software as a means of representing work and commerce can offer new insights on how systems of cultural meaning have been attached to, and expressed through, computers and computing.

Acknowledgments

This article draws on research that was supported by a Ruth Lilly Philanthropic Library Research Award from the University Library at Indiana University-Purdue University. I thank the University Library for supporting this research, and Bradley Fidler, Mary Gray, Lana Swartz, and Tarleton Gillespie for feedback on drafts of this article.

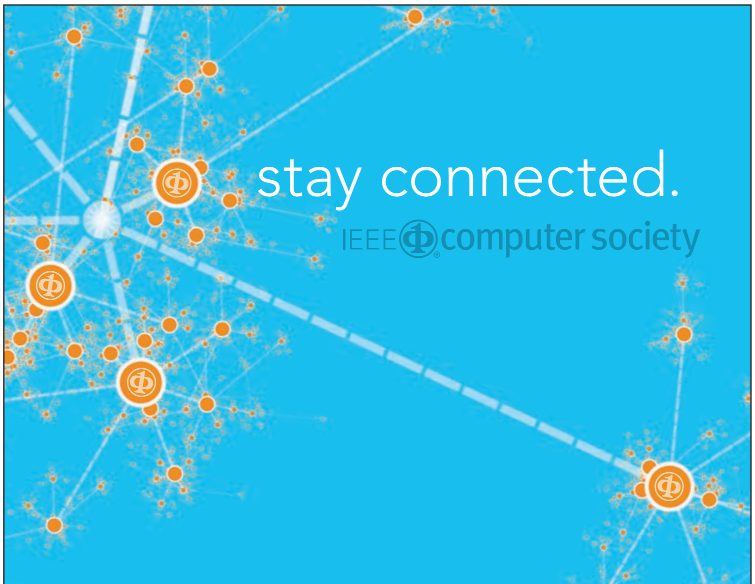
References

1. K. Flemke, to P.R. Ellis, correspondence, 23 Sept. 1982, Ruth Lilly Special Collections and Archives, IUPUI Univ. Library, Lilly Archives, Junior Achievement Records 1916–2002, Education Team Records Collection, box 89, folder 18.
2. "NBLC Afternoon Program Applied Economics Demonstration Script," 1982, Lilly Archives, Junior Achievement Records 1916–2002, Education Team Records Collection, box 51, folder 6.

3. C. Jack, "How Facts Survive: Sponsored Economic Education Media and American Social Imaginaries, 1974–1986," PhD dissertation, Dept. of Communication, Cornell Univ., 2016.
4. K. Phillips-Fein, *Invisible Hands: The Businessmen's Crusade against the New Deal*, W.W. Norton & Company, 2009.
5. L. Vinsel and A. Russell, "Hail the Maintainers," Aeon, 7 Apr. 2016; https://aeon.co/essays/innovation-is-overvalued-maintenance-often-matters-more.
6. T. Streeter, *The Net Effect: Romanticism, Capitalism, and the Internet*, New York Univ. Press, 2010; F. Turner, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*, Univ. of Chicago Press, 2008.
7. E. Medina, *Cybernetic Revolutionaries: Technology and Politics in Allende's Chile*, MIT Press, 2011; B. Peters, *How Not to Network a Nation: The Uneasy History of the Soviet Internet*, MIT Press, 2016.
8. I allude here to media theorist Stuart Hall's writings on ideology and media, in which Hall argues that ideology is produced through struggles over the meaning of language and symbols. According to Hall, media production provides social actors with an opportunity to decouple (or as he puts it, disarticulate) certain terms and symbols from their existing meanings and to articulate to those terms and symbols preferred meanings that advance different political projects. For further reference, see S. Hall, "The Rediscovery of 'Ideology': Return of the Repressed in Media Studies," *Culture, Society and the Media*, M. Gurevich et al., eds., Routledge, 2005.
9. J.J. Francomano, W. Lavitt, and D. Lavitt, *Junior Achievement, a History: A Personal Account of Managing Change from Drill Press to Computer*, Junior Achievement, 1988.
10. W.L. Norris and J.A. Seeley, "Partial Briefing Information on Initial Pilot Evaluation for Junior Achievement, Incorporated," 7 July 1983, Lilly Archives, Junior Achievement Records 1916–2002, Education Team Records, Collection, box 50, folder 25.
11. J. Tesch to P. Harder, "Student Company Computer Simulation," memo, 25 Mar. 1985, Lilly Archives, Junior Achievement Records 1916–2002, Education Team Records Collection, box 51, folder 8.
12. "Applied Economics Computerized Student Company Software User Manual, Experimental Edition," Junior Achievement, Sept. 1986, Lilly Archives, Junior Achievement Records 1916–2002, Education Team Records Collection, box 54, folder 16.
13. J. Rankin, "Toward a History of Social Computing: Children, Classrooms, Campuses, and Communities," *IEEE Annals of the History of Computing*, vol. 36, no. 2, 2014, pp. 86–88.

Caroline Jack *is a doctoral candidate in the Department of Communication at Cornell University. She will be a postdoctoral fellow at the Data & Society Research Institute starting in fall 2016. Contact her via www.cjack.org.*

This article originally appeared in
IEEE Annals of the History of Computing, *vol.38, no. 3, 2016*



EDITOR DAVID ALAN GRIER
George Washington University; grier@gwu.edu

This article originally appeared in Computer, *vol. 49, no. 8, 2016*

Changing Trust

David Alan Grier, George Washington University

Public-key cryptography is like a speed bump—it changes the nature of trust from a question of character to a mechanical reaction.

When trying to understand blockchains, bitcoins, and cybercurrency, it's very easy to get lost in the details of public-key cryptograpy and the power of peer-to-peer networking. However, the basic lesson of this technology is very simple: it's like a speed bump, in that it changes the nature of trust from a question of character to a mechanical reaction.

Speed bumps are intended to make drivers slow down for the safety of pedestrians and neighborhood children. However, most drivers reduce their speed out of concern for their car. When a speed bump is encountered, a child is a moral abstraction to drivers. But damaging their suspension or hitting their head on the car roof is a physical reminder.

Of course, blockchains are meant to provide trustworthy transactions, not to slow cars. They're a public, secure ledger, and they record transactions in a way that's very difficult to modify. With bitcoins, the blockchain records the creation of each coin and the movement of coins from one hand to the next. No one can create counterfeit bitcoins or spend coins they don't have.

See **www.computer.org/computer-multimedia** for multimedia content related to this article.

In changing the nature of trust, a blockchain destroys one set of institutions and creates new ones. It moves work from banks, courts, and governments to computer owners. Because it's a distributed database that resides on multiple computers, a blockchain record is valid if it's verified by a majority of computers that validate new records. "The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes," explains the original 2008 paper on bitcoins (https://bitcoin.org/bitcoin.pdf).

Over the past nine months, blockchain technology has faced a new crisis of trust. Australian engineer Craig Wright has identified himself as the inventor of blockchains and bitcoins, although the original papers were circulated under the pseudonym Satoshi Nakamoto. While preparing a portfolio of patents related to blockchain technology, Wright seemed to have decided that this work would be better received if it were identified with the inventor of bitcoins.

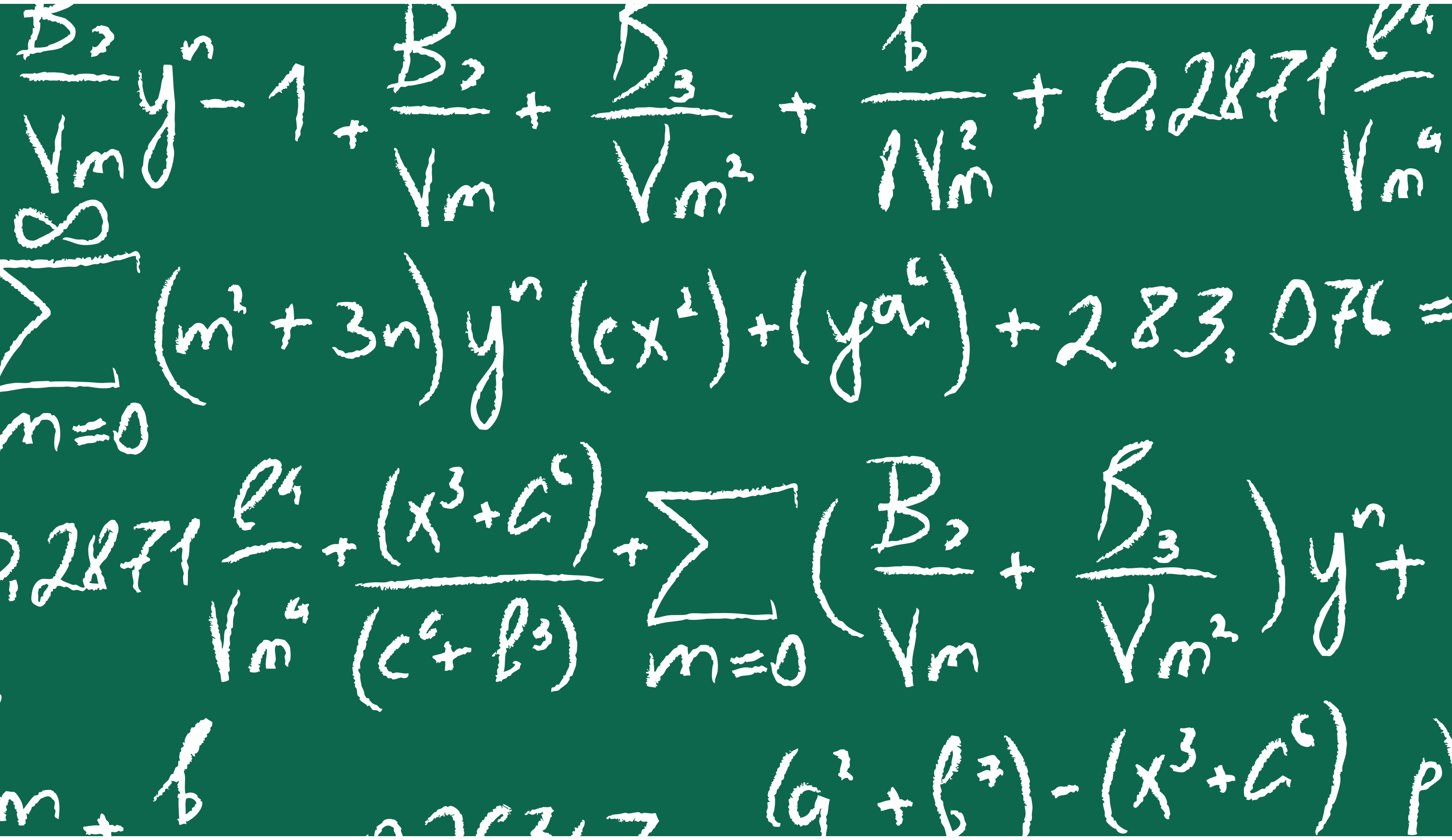
There are rules for assigning patents that are administered by national patent offices, one of the institutions charged with maintaining public trust. To obtain a patent, an individual must be the first to file a claim, have an invention that's novel, and demonstrate an idea that's not obvious to people working in the field. Although the original bitcoin and blockchain ideas aren't patentable, because they've been released to the public, other ideas associated with them might be.

However, the concept of inventor isn't always administered by an institution of trust. Instead, it's often bestowed by a community that's more interested in the technology. Wright has presented his story to the bitcoin community but has yet to convince enough members of the community that he's Satoshi Nakamoto. He's established that he knew some of the early workers on bitcoins and had access to some of the first encrypted records, but he took some actions that might suggest he's a fraud. (Andrew Hagan tells the story in detail in "The Satoshi Affair," *London Rev. of Books*, vol. 38 no. 13, 2016, pp. 7–28.)

As is common in Internet communities, the bitcoin community erupted into an angry and violent discussion about Wright. After a few days, Wright withdrew his claims. "I broke," he wrote. "I do not have the courage."

One of the purposes of trusted institutions is to remove emotion from decisions, to provide a systematic way of establishing trust. Patent offices, banks, and government agencies do it one way. Bitcoins and blockchains have shifted that process, for certain kinds of transactions, to an algorithm—one that relies on majority computing. It hasn't, however, solved all problems of trust, nor provided a way to convert a community's emotions into a sense of trustworthiness. C

DAVID ALAN GRIER is an associate professor at George Washington University. Contact him at grier@gwu.edu.



The Power to Create Chaos

Konrad Hinsen | Centre de Biophysique Moléculaire in Orléans

Occasionally, I hear from readers about the articles I write for the Scientific Programming department. My recent piece on technical debt[1] provoked more feedback than anything else I've written about here before, so I suspect the topic resonated with many of you. Most the emails I received were about personal experiences people have had with technical debt recognized too late to be handled gracefully. But a few concentrated on a very specific point: my claim that computers exhibit chaotic behavior, which is a problem when using them in research. I'll take these reactions as a pretext to elaborate on this point, which isn't (in my opinion) yet sufficiently appreciated.

I call this a pretext because I won't actually address the objections raised in much detail: they focus on the problems of defining chaos mathematically for discrete rather than continuous systems. The standard definitions of chaos refer to infinitesimal changes in initial conditions, which make no sense for computers and other discrete state systems because the smallest possible change is one bit. Readers interested in this topic can find detailed discussions in the literature on cellular automata; Stephen Wolfram's *A New Kind of Science*[2] is a good entry point, especially chapters 4 and 7. However, in the context of scientific programming, which is the topic of this department, the precise mathematical definition of chaotic behavior is much less relevant than its consequences on software development and testing.

Defining Chaos

Chaos is a mathematical concept from the theory of dynamical systems, which are defined by a state space and a time evolution rule for state. A dynamical system starts in an initial state; the time evolution rule is then applied to yield subsequent states. Both the state space and the time variable can be continuous or discrete. A computer is a dynamical system with a discrete state that consists of the computer's memory and its processor's internal state. Time is discrete as well, the elementary time step

2469-7087/16/$33.00 © 2016 IEEE

being the execution of one processor instruction. The processor's instruction set provides the details of the time evolution rule. A computation's initial conditions are memory contents plus processor state when the computation is started. Execution proceeds until the program reaches its end—if it ever does. The final memory contents contain the computation's result. Note that "memory" should be understood in a wide enough sense to include all data storage available, including hard disks, network storage, and so on. Note also that what I consider here is computation in the narrow sense of mechanically processing information. When you add multiple processes, communication between them, or external events, everything becomes more complicated.

The defining aspect of chaos is the strong sensitivity of a dynamical system's behavior to initial conditions: small changes in these conditions can cause large changes in the system's future behavior for which no useful bounds can be established. Chaotic behavior in nature makes the long-term evolution of many phenomena unpredictable even though they're deterministic. An often cited example is the weather, which can be predicted for only a very short time—about a week—not because of any inherently random processes or a lack of computational power but because the initial conditions that enter into the prediction can be measured only to some finite precision.

Computers are engineered dynamical systems for which these problems don't exist. We know a computation's initial state precisely—we can even store it for future reuse—but changes in the initial state are of interest as a way to explore the consequences of errors. Erratic behavior of the computing hardware itself is rare enough that it can safely be ignored, except for extremely large parallel computers. But human errors in the preparation of the initial state—program and input data—are an important cause of incorrect results. That's why it makes sense to ask whether a computation's behavior changes if the correct initial state is modified in some way.

Saying that a computer behaves chaotically means that a computation's result depends strongly on the initial state, to the point that a small change in this state can change the result beyond any useful predictable bound. As I already mentioned, some of the criteria of traditional chaos theory don't apply: changes can't be made infinitely small, as the smallest possible change is a one-bit flip, and deviations can't become infinite because the computation's state consists of a finite number of bits. The latter restriction applies to every physical system, of course, but the finiteness of our planet's atmosphere hasn't prevented scientists from applying chaos theory to weather forecasting.

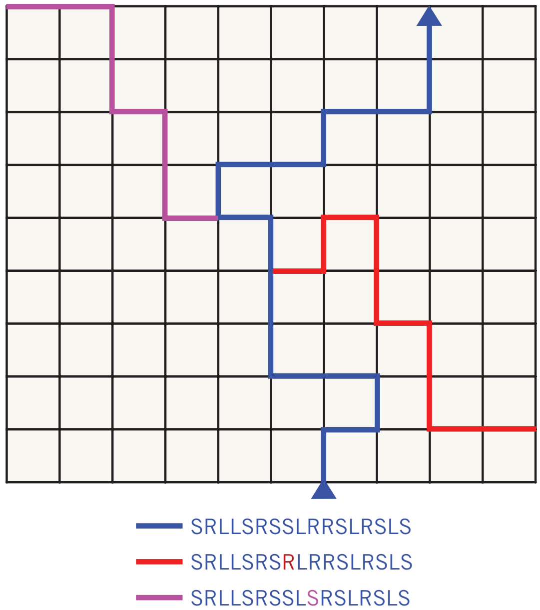


Figure 1. The consequences of minimal mistakes in driving directions. A list of driving directions for every corner—turn left (L), turn right (R), or continue straight on (S)—becomes a string made of the letters L, R, and S. The smallest possible mistake would be the replacement of a single letter. Two such minimally modified instruction sequences, in red and violet, show the resulting paths.

Amplifying Errors

The mechanism that causes chaotic behavior in computation is the amplification of small changes by subsequent steps. The impact of a one-bit flip can be small, say, if the bit happens to represent the least significant digit of an input number. At the other extreme, a one-bit flip in a processor instruction can crash the program, leading to no useful result at all. In between these two extremes, a one-bit flip can lead to results that differ from the correct result in arbitrary ways. The worst case isn't a program crash or a huge difference in the final result but an incorrect result that looks credible. Such a mistake has a good chance of going unnoticed.

I found a nice, simple illustration of error amplification in computation in a lecture by Gérard Berry (www.college-de-france.fr/site/gerard-berry/inaugural-lecture-2008-01-17-18h00.htm). Suppose you live in a city with a grid-like street layout (see Figure 1). You want to explain to a friend at the other end of the city how to reach you by car. You provide a list of driving directions that tell your friend what to do at every corner: turn left (L), turn right (R), or continue straight on (S). Your driving instructions are thus a string made of the letters L, R, and S, which isn't very different from a program written as a list of processor instructions. By following these instructions, your friend will move along the blue path from the bottom of the grid to the top.

But suppose that some mistake happens in transmitting the driving directions or that your friend takes a wrong turn. The smallest possible mistake would be the replacement of a single letter. Figure 1 shows two such minimally modified instruction sequences, in red and violet, together with the resulting paths. It's clear that the places you reach by following these modified instructions aren't at all close to the real destination or even close to each other. In fact, minimal mistakes can take you anywhere on the grid.

In this simple example, the mechanism of error amplification is easy to understand: each minimal one-letter change rotates the remaining part of the path by 90 degrees compared to the correct one. The more steps that remain to be done after the mutated instruction, the farther the arrival point will be from the intended one. This is also the general mechanism that makes a computation's outcome in the presence of mistakes so hard to predict.

An error that grows linearly with the number of computational steps isn't really that bad. It doesn't yet deserve the label "chaotic behavior." After all, our driving directions programming language provides several useful guarantees. No matter what mistake you make, a program consisting of N instructions is guaranteed to terminate after exactly N steps and yield a valid position on the grid (assuming the grid is big enough), which is at most $2N$ grid steps away from the intended destination. There's no possibility of nontermination (driving around the grid endlessly) or a crash (hitting an invalid position). Unfortunately, we don't get such guarantees for the programs we use in computational science. The reason why we have them for our driving directions is that the language is very limited; it can be processed by what's known as a *finite state machine* in automata theory, whereas standard programming languages require a more powerful automaton—in fact, the most powerful type of automaton known today—called a *Turing machine*. As in other aspects of life, more power also means more responsibility, as mistakes can have more serious consequences. Turing machines give you so much power that you can easily create chaos (www.scipublish.com/journals/AM/papers/1506). If we added features such as loops or tests based on current position to our driving directions language, we would need a Turing machine to process it, possibly condemning our friends, intentionally or by mistake, to spend the rest of their lives driving around the city.

In writing scientific software using Turing-complete languages, we have exactly that power and we should be constantly watching out for the unexpected consequences of mistakes. However, that isn't how computational scientists behave in practice. The dominant attitude is to trust computational results if they "look right," that is, they don't disagree with expectations and prior knowledge of the problem under study. I've even heard people say that they "trust their intuition" to spot potential mistakes in the results. Now that isn't in itself an absurd idea: experienced experimentalists do recognize suspect results coming from instruments they know well. It's certainly possible to develop an intuition for the credibility of data.

However, the crucial difference is that experimental equipment is carefully designed not to exhibit chaotic behavior, such that minor damage or production defects don't lead to unexpected results. If you put a mite under a microscope, a defect in the instrument could lead to a blurred image but not to an image showing an extra pair of legs. If the image is sharp enough to let you count the mite's legs, then you know you can trust that observation. But this approach doesn't carry over to software. There's no typical symptom of a programming mistake—anything is possible. Even the most experienced software developers can't judge if a program is correct by inspecting the source code, or by running it on a few sample inputs, except if the program is trivially small. Moreover, in many applications of computational science, correctness of a program isn't even something we could aim for. We can only decide if a program is correct if we have some other means of specifying what the correct result is. This often isn't the case in a research setting, where much code is written for computing quantities that nobody has ever computed before.

Mechanized Mathematics

A common attitude that we should probably start to question in this context is the one of considering computation as mechanized mathematics. The focus of mathematics is on precise statements that can be proven right or wrong. In research, we often use computation to explore imprecise statements—scientific hypotheses—whose domains of validity aren't known yet. A useful complementary notion to mathematical correctness is the robustness of the computational tools we use. This is well explained in an essay by Gerald Sussman that draws on his background in electrical engineering (http://groups.csail.mit.edu/mac/users/gjs/6.945/readings/robust-systems.pdf). A robust computer program produces reasonable output for reasonable input, even if the latter differs from what the program was initially designed to handle. We have similar concepts in computational science, for example, the notion of numerical stability of algorithms. But robustness criteria don't play a major role in the design and implementation of scientific software today.

To link these abstract considerations of potential chaotic behavior to the practice of computational science, I tried to apply them to my own work in molecular simulations. Most of my work is about extracting information from simulation trajectories, which are roughly 1-Gbyte datasets—too big for inspection by eye but small enough to be processed on my laptop. I spend much of my day writing, modifying, and running Python scripts that perform various geometrical and statistical analyses on these trajectories. These scripts are rather short but rely on a collection of libraries, ranging from general and widely used ones such as NumPy (www.numpy.org) or h5py (www.h5py.org) to domain-specific ones such as Mosaic (http://github.com/mosaic-data-model/mosaic-python).

The computation that's performed when I run one of my scripts is defined by the script, the file containing the trajectory, the Python language, and all the libraries used by the script. But that's just what's immediately visible. The libraries I've cited depend on other libraries, and the Python interpreter is written in C. A different way to present this complex assembly of software is as a set of consecutive layers that transform a general-purpose computer into a tool for performing a very specific analysis of a simulation trajectory. Each of these layers can be described in terms of the notation in which the additional information being added is expressed and the tool that processes this information:

1. the processor instruction set, executed by the computer;
2. the C language, translated into processor instructions by a C compiler;
3. the Python language, executed by an interpreter written in C;
4. the Python language augmented by the NumPy library, written in C and plain Python;
5. more libraries; and
6. the file format for the data files, interpreted by an analysis script written in Python and augmented by various libraries.

All the layers listed here define both a part of the computation and the notation in which some other part of the computation is expressed. For example, the Python language defines the data representation and memory management aspects of the computation, in addition to defining what is and isn't a legal Python program. My analysis script defines all the algorithms as well as the input file format for the trajectories.

For historical and practical reasons, we use different labels to refer to these notations: the first three are called *programming languages*, whereas the last one is a *file format*. The intermediate ones that simply add libraries are rarely recognized as distinct notations at all. To see that they really are, consider the small Python script:

```
import numpy
print(numpy.arange(5))
```

This isn't a valid program in the Python language, but it's valid in the Python-plus-NumPy language, which shows that these two languages are distinct. Libraries should thus be treated as language extensions.

All the layers but the last one are general-purpose, Turing-complete programming languages. There are, of course, significant differences between them, which is why these different layers exist. The processor instruction level isn't very convenient for human programmers, being difficult to read and understand. Moreover, processor instructions are a high-risk notation: any sequence of bytes can be executed as instructions, but most byte sequences won't produce anything useful and could even damage stored data. The C language is much more convenient and provides a better level of verification, with many possible mistakes caught by the compiler. The Python language is even more convenient and helps avoid mistakes by expressing a computation in much fewer lines of code. Similarly, each library layer adds more convenience for computations made up of the kind of operations that these libraries implement, at the same time preventing mistakes by allowing the programmer to write less code for a specific computation.

However, none of these layers adds useful guarantees about the computation's behavior. The potential of generating chaos remains present until the very last layer, which is defined by my trajectory analysis script. From the viewpoint of robustness, it would be preferable to have as much of the computation as possible expressed using notations that limit the impact of mistakes. Instead of libraries that add optional shorthand notation for some operations to a general-purpose language, we should have successive layers of languages that enforce the use of a more specialized and less dangerous notation.

This idea is sometimes advocated as the "principle of least power": every aspect of a computation should be described in a language with just as much expressive power as strictly required for the task (www.w3.org/DesignIssues/Principles.html). The additional guarantees that a limited language can make also offer more opportunities for analyzing and optimizing a program. Many domain-specific languages (DSLs) are based on this idea.[3] In contrast to a library, a DSL is both more and less than a general-purpose language, with the "less" part often including giving up Turing-completeness. A data file format is then nothing but an extreme case of a DSL in which only constant data is allowed and no algorithm can be expressed at all.

The main obstacle to such an approach is psychological. We associate the term "language" in the context of programming with something complex that takes many years to master. Most computational scientists would be happy to get away with learning only one programming language in their life. But as I explained earlier, a language can be a small variation on another one. Just as today's libraries are extensions to general-purpose languages, we could have library-like pieces of code that remove or restrict features of languages. As an example, removing while-loops and recursive function calls would turn Python into a language in which every program is guaranteed to terminate. A quick inspection of the trajectory analysis scripts that I wrote over the past few months showed that they could all be written in such a restricted Python dialect. In fact, they already are. I rarely need all the power that Python gives me. But I can't ask Python to verify that I didn't use that power by mistake.

These observations apply with minor variations in languages, libraries, and file formats to most computations done today in science and engineering. In the short history of computing, we can observe a general tendency toward providing more expressive power wherever some aspect of a computation is defined. In the early days, we had Fortran programs reading simply structured input files, but as soon as computers could handle larger programs, embedded scripting languages—usually Turing-complete—became a desirable feature for customizing application software. From there, it was a small step to writing application software in a scripting language augmented by DSLs. There are even cases of languages that became accidentally Turing-complete as features were added, the best-known example being C++ templates. It is, in fact, possible to do arbitrary computations, including chaotic ones, as part of the compilation of a C++ program. The downsides of too much freedom in program structure have been known for a while, leading to the widespread adoption of structured programming in the 1980s and to the growing popularity of functional programming in recent years.[4] But advocates of these approaches to safer programming were always keen to point out that no expressive power is lost in adopting them. Perhaps it's time to reevaluate the importance of computational omnipotence.

References

1. K. Hinsen, "Technical Debt in Computational Science," *Computing in Science & Eng.*, vol. 17, no. 6, 2015, pp. 103–103.
2. S. Wolfram, *A New Kind of Science*, Wolfram Media, 2002.
3. K. Hinsen, "A Glimpse of the Future of Scientific Programming," *Computing in Science & Eng.*, vol. 15, no. 1, 2013, pp. 84–88.
4. K. Hinsen, "The Promises of Functional Programming," *Computing in Science & Eng.*, vol. 11, no. 4, 2009, pp. 86–90.

Konrad Hinsen is a researcher at the Centre de Biophysique Moléculaire in Orléans (France) and at the Synchrotron Soleil in Saint Aubin (France). His research interests include protein structure and dynamics and scientific computing. Hinsen has a PhD in theoretical physics from RWTH Aachen University (Germany). Contact him at konrad.hinsen@cnrs.fr.

This article originally appeared in Computing in Science & Engineering*, vol. 18, no. 4, 2016.*

Finding the Emerging Technology Job You Want

Technologies change and advance rapidly. As a result, important new technologies emerge regularly. Finding jobs in these emerging fields is the focus of this month's careers article, which features interviews with three key employees of Hewlett Packard's HP Labs. R. Stanley Williams leads the research group's Foundational Technologies Group, which creates the mathematical and physical bases for new information ecosystems' technologies. Sharad Singhal is an HP distinguished scientist with research interests in speech and video processing, middleware, and large-scale distributed computing. Kirk Bresniker is the labs' chief architect of systems research, where he guides research and advanced development of novel hardware and software system designs. They co-authored the article "Adapting to Thrive in a New Economy of Memory Abundance," which appeared in *Computer*'s December 2015 issue.

ComputingEdge: What careers in the newest emerging technologies will see the most growth in the next several years?

Williams: In information technology, I see neuromorphic computing as offering the biggest opportunity for growth the next decade or two. Adapting concepts from the brain for computation will be the key enabler for robotics, the Internet of Things, autonomous vehicles, search, and learning.

Singhal: Technology is evolving at an ever-increasing pace, with the most recent advances in the areas of autonomous systems, such as self-driving cars, drones, and the Internet of Things. The combination of machine learning and the analytics of very large datasets with real-time control will provide many new opportunities in the next several years.

Bresniker: We're moving toward a future in which we create information systems capable of not just providing us hindsight (what has been happening) but also of giving us insight (what is happening right now) and eventually foresight (what is most likely to happen). Key technology trends here include the creation of intelligent, distributed systems—such as those that will be part

Published by the IEEE Computer Society

of the Internet of Things—that can house large amounts of information outside of traditional datacenters.

ComputingEdge: What would you tell college students to give them an advantage over the competition?

Williams: First, love to learn and learn how to learn. All technology fields are advancing exponentially, so whatever you know now will be obsolete in a decade. Build a foundation of basics, especially math, but be both willing and eager to learn new things. If that doesn't appeal to you, technology is the wrong place to be. Second, the most important skill for any technologist is communication. It doesn't matter how good your work is if you can't describe it to others in terms they can understand. Take an English composition course or other writing course. Get involved in an activity that will help you with spoken communication in English such as debate, drama, or journalism. And learn how to listen. Communications skills will be more important to your career in the long run than anything you will learn in any single science or engineering course.

Singhal: It's unlikely that you'll continue to do what you learn in college over your entire career. Indeed, technology careers involve a lifetime of learning and developing and demonstrating the ability to adapt to new information and tackle new problems.

Bresniker: There's much to learn that will not appear in your engineering or science curriculum. There's the legal stuff, such as what it's like to sit in a deposition across the table from an [intellectual property] litigator or how to read the unfamiliar language of a patent application to make sure your ideas are still in there. There are also the soft skills. How do you communicate when you're trying to get nontechnical people to understand something that only you know; something that even your fellow technologists don't understand yet? While you're in school, take advantage of the fact that you're on the same campus as people who are as passionate about arts, communication, law, and commerce as you are about technology. If there are cross-disciplinary or entrepreneurial programs available, take advantage of them. If there are internship programs, join them to get practical technical experience. But once you're in, ask about the bigger picture.

ComputingEdge: What advice would you give people changing careers midstream?

Williams: If you find you're not having fun at what you're doing, it's time for a change. You can only be really good at something if you enjoy it, and if you're not really good at what you're doing, your career will spiral downward fast. A major career change can jumpstart your outlook on life and provide excitement as you adapt to a new environment, even if it means a significant salary decrease. Value the accumulation of experiences rather than money.

Singhal: By mid-career, most of us tend to settle down into a routine, in which we're perceived as experts in a field, have a reasonably well-developed network, and can do our jobs effectively. However, if a job becomes just a job and is no longer fun, it's time to consider something different. Moving to a different field doesn't mean giving up on previous experience. Mid-career hires can bring valuable perspectives to a new job. However, while changing careers can be rejuvenating, it also requires accepting that you will be starting fresh and will make mistakes.

Bresniker: I've been at [Hewlett-Packard] for close to 27 years, so I may not the best one to answer that question. However, I have had many roles there. My best advice is to own your own career trajectory. There will always be a tension between you progressing in your arc and being a key contributor at your current level. Your manager loves you right where you are because you're helping her achieve her goals and she can count on you. But if you're ready to move up, then you need to speak up and take steps to accomplish it.

ComputingEdge: What do you consider to be the best strategies for professional networking?

Williams: Be a good listener. In discussions with people, respect their ideas even if you don't agree with them. If you can accept someone else's idea and give it back a little improved, you will have won an ally. If you're really good at what you do, people will come to you in droves. There will be no need to spend time with conventional networking. Seek out one or two mentors and repay their time spent with you by building their legacy. Then be a good mentor to a few select people who are able to benefit from what you offer. They will be your legacy.

Singhal: Professional networking is more than just accepting invites to LinkedIn or keeping up with people who can advance your career. It's about developing a group of peers you can reach out to but, more importantly, who are comfortable reaching out to you for technical or professional conversations. One of the best strategies I've found for networking is taking the time to help others either through mentoring or simply listening and answering a few questions or providing opinions when needed.

Bresniker: Don't stop at the traditional angles. Join IEEE, work your university alumni pool, use LinkedIn, but don't stop there. If you can code, find an open source project that is interesting to you and contribute. If you can't code, you can contribute to testing, documentation, localization, or marketing. Open source rewards the investment of sweat equity, it's more diverse and global than almost any other technical community, and it'll teach you to collaborate. It can be a bit intimidating. There are big ideas and big egos in the open source community, but that's part of the energy and passion that brings people together.

ComputingEdge: What should applicants keep in mind when applying for emerging technology jobs?

Williams: The people interviewing you will probably know a lot more than you do about an emerging field, no matter how good your background is, so be humble. Demonstrate a strong interest in the field and a willingness to learn.

Singhal: Emerging technology jobs are, by nature, concerned with unproven ideas and directions. The job is likely to be exciting but also might lack structure and well-defined tasks. It will require flexibility and the ability to switch to a new way of thinking or of moving from one problem to the next. An attitude that anticipates and thrives on change will help, as will being receptive to ideas from other disciplines.

Bresniker: Emerging technology is unproven, so you're taking a risk. You're investing part of your career trajectory in something uncertain. There is more potential upside, but there is also going to be less structure, fewer rules, and more independence. That can be enervating or unsettling. You need to know your own personality and comfort around uncertainty. You might well be working on small teams, which offer a unique opportunity to cross-train in various roles. I had that experience when I was the only electrical engineer on a small tiger team charged with creating one of the first low-cost UNIX server designs. That provided me with the breadth of knowledge to later take on chief technologist roles.

ComputingEdge's Lori Cameron interviewed Williams, Singhal, and Bresniker for this article. Contact her at l.cameron@computer.org if you would like to contribute to a future *ComputingEdge* article on computing careers. Contact Williams at stan.williams@hpe.com, Singhal at sharad.singhal@hpe.com, and Bresniker at kirk.bresniker@hpe.com.

Attack Surfaces

Steven M. Bellovin
Columbia University

Attack surface is one of those core concepts that never gets the attention it deserves. It should, because properly understood, attack surface not only helps people analyze system designs, but also explains why some system changes help and others hinder.

Roughly speaking, a system's attack surface is the set of ways that it might be susceptible to an attack. A program that's listening on three network sockets has a larger attack surface than one that's listening on only two of those sockets. It's not necessarily more vulnerable—assuming there are no vulnerabilities present in the code listening on the third one—but it's one more thing to analyze and, if necessary, secure.

Obviously, not all elements of the attack surface are equally risky. To give the simplest example, a service accessible only from a local network is less problematic than one accessible from the entire Internet. Attack surface elements can thus be weighted according to their riskiness. However, we have to look broadly at all system elements, including the user community, when assessing the attack surface.

The nature of looking broadly is well-illustrated by this quote from Paul A. Karger and Roger R. Schell's classic Multics security evaluation: "Trap doors can be inserted during the distribution phase. If updates are sent via insecure communications—either US mail or insecure telecommunications, the penetrator can intercept the update and subtly modify it. The penetrator could also generate his own updates and distribute them using forged stationery." In other words, the update process might be part of the attack surface, too, if your adversaries have sufficient resources to attempt this.

The concept of attack surface also explains why firewalls were originally so important—and why they don't work as well today. In 1994, Bill Cheswick and I wrote: "The biggest single reason that a firewall is likely to be more secure is simply that it is not a general-purpose host. Thus, features that are of doubtful security but add greatly to user convenience—NIS [Network Information Service], rlogin, etc.—are not necessary. For that matter, many features of unknown security can be omitted if they are irrelevant to the firewall's functionality." We were really saying that a firewall has a smaller attack surface than a standard computer. For that matter, a wide-open network has an attack surface that's roughly the sum of the attack surfaces of every computer, network link, user, and so on, in the entire organization. Today, though, a corporate firewall proxies or passes through a lot more traffic. Users' mailers and Web browsers have more functionality. Smartphones live inside and outside the firewall. In other words, the attack surface of the enterprise (and perhaps of the firewall itself) is vastly larger.

We can use this concept to understand the current debate over encryption and, in particular, why most cryptographers and system security specialists oppose provisions for "exceptional access." Quite simply, we know from experience that cryptographic protocols and their implementations have a high weighted attack surface: especially for their size, they're very hard to get right. It also becomes clear why an encrypted smartphone is less secure than a computer with full-drive encryption: in the latter, nothing is running at boot time except for the decryption code. With phones, however, much of the system functionality (often including the ability to receive SMS messages) is available even when the decryption key hasn't been supplied. Translation: the attack surface is much larger.

These insights are useful, but it's important not to rely too much on attack surface analysis to the exclusion of other measures. It's all too easy to underestimate the weight that should be assigned to any given component, or to miss some components entirely. Just because a component doesn't seem risky doesn't mean that's actually the case: security bugs can happen anywhere. ■

Steven M. Bellovin is a professor of computer science at Columbia University. Contact him via www.cs.columbia.edu/~smb.

This article originally appeared in IEEE Security & Privacy, *vol. 14, no. 3, 2016*

SENIOR RESEARCHER, Warren, MI, General Motors. Design &build Advanced Vehicle Qlty Analytics Enterprise apps in Primefaces with Spring, Java Server Faces, AJAX, EXT JS &Java Script. Design data models, perform the ETL jobs using DataStage, nCluster Loader, SQL Loader &Enterprise Guide Client tools, performing cold-backups, administering databases in Teradata Aster &Oracle using TOAD. Program in Java, J2EE with Spring Framework, C++, SAS Soft, &Web Services using SOAP, WSDL &UDDI to deploy apps in Weblogic server. Master, Computer Science, Mathematics, Engineering or Operations Research. 1 yr exp as Programmer Analyst, Consultant or Engineer in job offered. Mail resume to Alicia Scott-Wears, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI 48265, Ref#408.

SYSTEM ENGINEER, General Motors, Detroit, MI. Analyze &test Hands Free Calling monitoring OnStar sys performance using Java-based OnStar NA, Europe &Mexico systems, to maintain highest possible svce delivery rates &validate adequate sys reqs (using cogons &Oracle) are documented for changes to sys platforms by maintaining close working relationship with IT, Service Delivery, &Engrg depts. Apply knowledge of vehicle infotainment modules. Ensure technology-vehicle connectivity incldg OnStar sys, ignition sys, roadside assistance, emergency svces &packages. Identify systemic issues affecting sys performance &create strategies to improve performance. Evaluate Key Performance Indicators (KPIs), design &dvlp performance metrics that represent End to End (E2E) svce delivery performance. Assist IT sustain team in identifying &troubleshooting production issues for U.S. &Europe systems. Utilize support tools to resubmit errors caused by production incidents to mitigate impact to customers. Select appropriate regression test cases to be run for SIT/UAT end to end performance, review test results, &write new test cases if necessary for new content, using QTP, ALM &OnStar TDM data tools. Master, Information Technology, Engineering or related. Required exp must include QTP soft &36 months as Test Lead, Test Engineer, System Engineer or related preparing Test Scenarios &Test Cases for vehicle infotainment modules such as Onboarding the Customer, Hands Free Calling, Ignition Block, Immediate Theft, Stolen Vehicle Assistance, Connectivity Package and/or Data Plans, based on client specs in OnStar domain. Will accept bachelor's or foreign equiv degree in Information Technology, Mechanical or Automotive Engineering or related, followed by at least 5 yrs progressive exp in specialty, in lieu of required education &exp. Will also accept any equally suitable combination of education, training, and/or exp which would qualify applicant to perform job offered. Mail resume to Alicia Scott-Wears, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI 48265, Ref#2375.

SR. APP DEVELOPER: complex code design; TIBCO development & integration; design, test, implement. of tech. solutions & integration patters; mentoring in coding, design, integration & Agile develop. methodologies. MS in CS, EE or related plus 2 ys of exp. or BS+5 yrs. Job in Columbia, MD. Email: sarah.ulep@laureate.net w/ Job #15944BR in subj line. Laureate Education, Inc. 650 S Exeter St. Balto. MD 21202. EOE.

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST - SUPPORT

positions in **Lehi, UT**.

Job duties include: Deliver post-sales support and solutions to the Oracle customer base while serving as an advocate for customer needs. Research, resolve, and respond to complex customer inquiries and technical questions regarding the use of, and troubleshooting for, our product line. May telecommute from home.

Apply by e-mailing resume to peter.zis@oracle.com, referencing 385.18415.

Oracle supports workforce diversity.

TECHNOLOGY

Oracle America, Inc.

has openings for

SYSTEMS ANALYST - SUPPORT

positions in **Frisco, TX**.

Job duties include: Deliver post-sales support and solutions to Oracle customer base while serving as an advocate for customer needs. Travel to various, unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to ely.kahwaty@oracle.com, referencing 385.15735.

Oracle supports workforce diversity.

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Culver City, CA**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to neal.pollack@oracle.com, referencing 385.19113.

Oracle supports workforce diversity.

VANDERBILT UNIVERSITY

THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE (EECS) AT VANDERBILT UNIVERSITY is seeking candidates for two tenured/tenure-track (T/TK) faculty positions. Appointments at all ranks will be considered; we prefer to fill at least one position at the assistant professor of computer science rank. Successful candidates are expected to teach at the undergraduate and graduate levels and to develop and grow vigorous programs of externally funded research. Areas of focus for this search are: (1) **Computer Assisted Surgery and Interventions.** We seek world-class expertise in medical image and signal analysis, computer vision, and medical robotics. The successful candidate will expand the CS curriculum in these areas and have a strong interest in translational research, collaboration with the Vanderbilt University Medical Center, and will be expected to engage with the Vanderbilt Institute in Surgery and Engineering (VISE: http://www.vanderbilt.edu/vise). VISE's mission is the creation, development, implementation, clinical evaluation and translation of methods, devices, algorithms, and systems designed to facilitate surgical and interventional processes and their outcome. (2) **Big Data/Data Science/AI.** We seek world-class expertise in broadly defined areas of data science, machine learning, data mining, visualization, computer vision, and/or artificial intelligence. The Vanderbilt CS program provides a unique, collaborative, and interdisciplinary research environment. New trans-institutional programs are creating opportunities for research on issues of broad significance that create and extend collaborations across multiple fields.

Vanderbilt University is a private, internationally renowned research university located in vibrant Nashville, Tennessee. Its 10 schools share a single cohesive campus that nurtures interdisciplinary activities. The School of Engineering is on a strong upward trajectory in national and international stature and prominence, and has built infrastructure to support a significant expansion in faculty size. In the 2015 rankings of graduate engineering programs by U.S. News & World Report, the School ranks third among programs with fewer than 100 faculty members. 5-year average T/TK faculty funding in the EECS Department is nearly $1M per year. All junior faculty members hired during the past 15 years have received prestigious young investigator awards, such as NSF CAREER and DARPA CSSG.

With a metro population of approximately 1.5 million people, Nashville has been named one of the 15 best U.S. cities for work and family by Fortune magazine, was ranked as the #1 most popular U.S. city for corporate relocations by Expansion Management magazine, and was named by Forbes magazine as one of the 25 cities most likely to have the country's highest job growth over the coming five years. Major industries include tourism, printing and publishing, manufacturing technology, music production, higher education, finance, insurance, automobile production and health care management.

Vanderbilt University is an equal-opportunity, affirmative-action employer that aspires to become a leader among peer institutions in making meaningful and lasting progress in responding to the needs and concerns of women and members of under-represented minority groups. Applications should be submitted on-line at: https://academicjobsonline.org/ajo/jobs/7736. For more information, please visit our web site: http://engineering.vanderbilt.edu/eecs. Applications will be reviewed on a rolling basis beginning November 1, 2016 with telephone interviews beginning December 1, 2016. The final application deadline is January 15, 2017.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Software Engineer: plan, design & implement functional, system, & regression tests within an automation test framework. Mail resume w/job code #37185 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Software Engineer: Design & implement large distributed systems that scale well to petabytes of data and tens of thousands of nodes. Mail resume w/job code #36109 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

CLOUDERA, INC. is recruiting for our San Francisco, CA office: Software Developer: Design & implement file & data formats such as Parquet & Avro for big data processing. Define and implement binary serialization & access methods in different program languages. Mail resume w/ job code #35974 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

SENIOR SOFTWARE ENGINEER: (Multiple Positions Available). Full lifecycle development. Designing, coding & debugging applications in various software languages such as C#, SQL, Web Services, Use of .NET including the CLR, LINQ & multithreaded programming. Software analysis, code analysis, requirements analysis, software review. OOA & OOD. Supervise the maintenance of system by lower level engineers etc. Master's degree in Computer Science or Computer Information System or alternatively Bachelor's Degree in Computer Science or Computer Information Systems with 5 years of experience. 40 hr/wk. Job Site & Intvu: Marina Del Rey, CA. Send resume to Solid Commerce att: Alon Berkovich at alon.b@solidcommerce.com.

IT PROFESSIONALS: Established IT firm has multiple openings for IT Project Managers, Engagement Managers, Technical Project Leads, Technical Team Leads, Account Directors, Business Analysts (Systems), Sr. Consultant/Software Engineers & Software Developers. PROJ. MANAGERS require Master's or equiv. in Engg (any), CS, IS or related & 12 mos' relevant indus exp. ENGAGEMENT MANAGERS require Master's or equiv. in Business Administration or Mgmt, CS, IT, Engg (any) or related & 12 mos' relevant indus exp. PROJ. & TEAM LEADS require Master's or equiv. in Engg (any), CS, Computer Applications or related & 12 mos' relevant indus exp. For Managers & Leads, we also will accept a Bachelor's or equiv. in the fields stated & 5 yrs' progressively responsible

& relevant indus exp. ACCOUNT DIRECTORS require a Bachelor's or equiv. in IT, Info. Systems, Mgmt, Finance or related & 5 yrs' relevant indus exp. BUSINESS ANALYSTS (SYSTEMS) require a Bachelor's or equiv. in Business Administrating or Mgmt, CS, IT, Engg (any) or related & 24 mos' relevant indus exp. SR. CONSULTANT/SOFTWARE ENGINEERS & SOFTWARE DEVELOPERS require a Bachelor's or equiv. in Engg (any), CS, IS or related & 24 mos' relevant indus exp. ALL positions based out of Edison, NJ HQ & subject to relocation to various unanticipated locations throughout the U.S. Qualified applicants mail resumes to: HR Manager, SYSTIME Computer Corporation (dba KPIT), 379 Thornall Street, Edison, NJ 08837.

COMPUTERS: Tech Mahindra (Americas) Inc. is seeking to fill numerous IT positions. Prgrm Mgrs to oversee & manage mult. IT projects, proj. planning, dvlpmnt, implementation, acct & delivery mgmt; Proj. mgrs to oversee & manage IT teams w/dvlpmnt of various sftwre apps. Sys. Analyst/Programmers/ Quality & Tech Architect/ Tech Solutions Architect /Quality Engg /Test Eng/ Sftwre Eng. / sftwre Developer / Systems Admin/DBAs/DB Architect /Network Engr/Network Admin/ to analyze, design, dvlp, test & maintain comp software apps, systems, databases, or networks through all phases of sftwre dvlpmt life cycle (Sftwre Eng. may also lead a team on various projects); Telecom solutions Architect / Application Architects/Electrical Eng to use various technologies to dsgn & dvlop telecom software apps. Mech. Eng (CAD/CAM) to design, develop, validate & perform structural calculations, product improvement, & provide tech. support to design teams at high levels utilizing specific mech. tools. Sales Eng/Bus. Analyst/ Mgmt Analyst for solutions/pre-sales activities w/relev. industry experience. IT Bus. Dev. Mgrs to create new business, negotiate contracts & dvlp proposals for customized IT solutions. Rel. Mgrs to manage/outsource commercial IT/Eng. deals, monitor & maintain existing accts. All Tech/Mgrial. positions require a MS or BS degree or foreign equiv. in CS, Comp Apps, CIS, IT, Eng, Bus mgmt/admin or closely related fields and relevant industry exp. Sales/Rel.Mgrs. require a MS or BS degree or foreign equiv. in Bus. Admin, Eng. or closely related field & relev. Industry exp. Positions are based out of corp. HQ in 4965 Preston Park Blvd, # 500, Plano, TX 75093 & subject to travel & relocation to client sites located throughout the U.S. Mail resume & position applied for w/JOB CODE: 091E16 to Visa Cell, Tech Mahindra (Americas) Inc., 1001 Durham Avenue, Suite 101, South Plainfield NJ 07080.

BAYLOR UNIVERSITY 1845

Baylor University is a private Christian university and a nationally ranked research institution, consistently listed with highest honors among *The Chronicle of Higher Education*'s "Great Colleges to Work For." Chartered in 1845 by the Republic of Texas through the efforts of Baptist pioneers, Baylor is the oldest continuously operating university in Texas. The university provides a vibrant campus community for over 15,000 students from all 50 states and more than 80 countries by blending interdisciplinary research with an international reputation for educational excellence and a faculty commitment to teaching and scholarship. Baylor is actively recruiting new faculty with a strong commitment to the classroom and an equally strong commitment to discovering new knowledge as we pursue our bold vision, Pro Futuris. (www.baylor.edu/profuturis/).

FACULTY POSITION

Assistant, Associate or Full Professor of Computer Science in the area of Software Engineering

The Department of Computer Science at Baylor University seeks a productive scholar and dedicated teacher for a tenured or tenure-track position beginning August 2017. Viable candidates must have a PhD in Computer Science or a closely related field, demonstrate scholarly capability and an established and active independent research agenda in software engineering and related areas. The ideal candidate will also have leadership experience, a commitment to undergraduate and graduate education, effective communication and organization skills and a strong research record that includes significant external funding. Qualities of a successful candidate for a senior position include leadership experience and a strong record of an independently funded agenda. All candidates are expected to exhibit a passion for teaching and mentoring at the graduate and undergraduate level.

The Department offers a B.S. in Computer Science degree, a B.A. degree with a major in Computer Science, a B.S. in Informatics degree with a major in Bioinformatics, a B.S. in Computer Science with a major in Computer Science Fellows and a M.S. degree in Computer Science and a Ph.D. degree in Computer Science, and is rapidly expanding with its new Ph.D. program.

Applications will be accepted until the position is filled. To ensure full consideration, complete applications must be submitted by **12/01/2016**.

APPLICATION PROCEDURE: Please submit 1) a letter of application, which includes the applicant's anticipated rank, 2) a current curriculum vitae, 3) a statement of teaching interests, 4) a statement of research plans related to Baylor's programs, 5) transcripts, 6) the names, addresses, and phone numbers of three individuals from whom you have requested letters of recommendation to: Dr. Eunjee Song (Search Committee Chair), Baylor University, One Bear Place #97141, Waco, Texas 76798-7141.

In addition, as an openly Christian institution, Baylor seeks to establish an intentionally Christian environment, where students and faculty are encouraged to pursue both faith and reason. In order to achieve this goal, we ask that applicants submit a brief statement addressing their Christian commitment. This may include church or religious affiliation, active membership in Christian organizations, or a personal statement.

Materials may be submitted electronically to Dr. Eunjee Song at eunjee_song@baylor.edu. Please combine all submitted material into a single pdf file.

To learn more about the above position, the Department of Computer Science, the School of Engineering and Computer Science, and Baylor University, please visit the appropriate URL:
https://jobs.baylor.edu/postings/1276;
http://www.ecs.baylor.edu/computerscience;
http://www.ecs.baylor.edu;
or http://www.baylor.edu.

Baylor University is a private not-for-profit university affiliated with the Baptist General Convention of Texas. As an Affirmative Action/Equal Opportunity employer, Baylor is committed to compliance with all applicable anti-discrimination laws, including those regarding age, race, color, sex, national origin, marital status, pregnancy status, military service, genetic information, and disability. As a religious educational institution, Baylor is lawfully permitted to consider an applicant's religion as a selection criterion. Baylor encourages women, minorities, veterans and individuals with disabilities to apply.

Juniper Networks

Juniper Networks is recruiting for our Sunnyvale, CA office:

Technical Systems Analyst #36347: Provide applications development and enhancement functions, including analysis, design, coding and testing using the ABAP Development Workbench.

Software Engineering Sr. Manager #8204: Manage a team of customer escalation engineers to determine root-causes and corrective actions to address systemic issues with Juniper switches.

Software Engineer Staff #12096: Design, develop, troubleshoot & debug requirements of software features related to routing protocols and services in Layer 2 & Layer 3 Virtual Private Networks (VPNs) & Virtual Private LAN Service (VPLS).

Build/Release Engineer #37580: Work with release management, integration engineers, developers, managers, project managers on Release build issues. Perform daily and release builds and triage issues.

Test Engineer #34470: Review design and functional specifications and develop test plans. Execute the testing for Layer 2 and Layer 3 protocols. Execute the testing for new hardware platforms.

Product Manager #22015: Drive software requirements for routing and switching products across Juniper. Work with customers and internal teams to translate complex requirements into engineering product requirements.

Mail single-sided resume
with job code # to
Juniper Networks
Attn: MS A.4.435
1133 Innovation Way
Sunnyvale, CA 94089

Computer Science Faculty Positions

Openings exist for multiple tenure-track faculty positions at the Assistant or Associate level to begin August 2017. Successful applicants must demonstrate the potential to contribute to the University of Alabama's initiatives with respect to water, the Alabama Water Institute (http://awi.ua.edu/), and/or transportation, the Alabama Transportation Institute (http://ati.ua.edu/). This includes, but is not limited to, areas such as big data, spatial data, data analytics, data visualization, machine learning, vehicular networks, software modeling, software engineering, security, robotics, and autonomous vehicles.

The Computer Science Department has 22 faculty members (14 tenured/tenure-track faculty), over 600 undergraduates in an ABET-accredited program, and approximately 50 graduate students. The Department generated over $13 million in research expenditures in FY 2015 and our doctoral program has produced 33 graduates in the past five years.

Applicants should apply online at http://facultyjobs.ua.edu/postings/39546. Applicants must have an earned doctorate (Ph.D.) in computer science or a closely related field. For additional details, please contact Dr. David Cordes (faculty.search@cs.ua.edu) or visit http://cs.ua.edu. The University of Alabama is an equal opportunity/affirmative action employer. Women and minority applicants are particularly encouraged to apply.

Fiberutilities Group, located in IA, is searching for an Integrated Network Administrator. Job duties: Evaluate, plan, install, configure and coordinate implementations of voice and data network communications products and services (WAN, voice and data LAN, WLAN, Firewalls, VPN's, VOIP, PBX). Using leading network hardware and software platforms to include, Cisco, Juniper, Ciena, Avaya, and Microsoft, design solutions, prepare detailed network design documents that articulate technical, budgetary and temporal specifications for proposed solutions to address client problems.

Bachelor's Degree in Computer Science, Information Technology or Computer Information Systems required plus 5 years experience working as a Network Engineer or Administrator in an enterprise setting with exposure to both voice and data technologies. Must have at least one active certification in the network administration discipline (e.g. Cisco, CCNA or similar). Will accept 3 year degree in Computer Science, Information Technology or Computer Information Systems. Candidate will be required to travel and relocate to various unanticipated client locations in the U.S. Current worksites are Cedar Rapids, IA and Peoria, IL. To apply, email your resume to msobolik@skywalkgroup.com.

SOFTWARE DEVELOPER - design, develop, test & implement application s/w utilizing knowledge of SAP BW, SAP ECC, Business objects, SAP HANA, MS Sql Server, SAP Analytics/Cloud Analytics, SAP HCP, BEx Analyzer, BEx Query Designer, Report Designer, SAP Cloud Analytics, SAP Cloud Application studio, Crystal Reports, Unix & Windows. Must be willing to travel & reloc to unanticipated client locations throughout the US. Reqs MS in comp sci, eng or rel. Mail resumes to Globalsoft Solutions Inc. 505 Thornall Street, Suite # 300, Edison, NJ 08837.

PROGRAMMER ANALYST. IT company in Matawan, NJ seeks Programmer Analyst to analyze, design, test, implement web based, client/server applications utilizing knowledge of C++,ASP.NET, C#.NET, ADO.NET, SQL Server, Oracle SQL Developer Web Services, WCF, HTML, JavaScript, XML, VBScript, JQuery, Knockout, VS, .NET, NUnit, Manual, MSTest and Windows. Interact with clients to gather requirements, prepare technical and functional specifications, create various types of tests, troubleshoot, maintain and fine tune applications. Apply with 2 copies of resume to HR, Questsoft Solutions Inc, 432 State Rt.34, Ste#3A, Matawan, NJ07747 orimmi@questsoftsol.com.

JUNIPER NETWORKS is recruiting for our Sunnyvale, CA office: Technical Support Engineer #16500: Deliver in-depth diagnostics & root-cause analysis for network impacting issues on Juniper's routing products to large ISP & enterprise customers. Technical Support Engineer #38980: Deliver technical support for complex network-impacting issues on Company's routing products (Internet backbone routers) to large Internet Service Provider and/or Enterprise customers. Software Engineer Staff #38534: Design, implement, and support software features and solutions based on Layer 2 and Layer 3 technologies for Juniper's networking products. Mail single-sided resume to Juniper Networks, Attn: MS A.4.435, 1133 Innovation Way, Sunnyvale, CA 94089.

CLASSIFIED LINE AD SUBMISSION DETAILS:

Rates are $425.00 per column inch ($640 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to:

DEBBIE SIMS
Classified Advertising
Email: dsims@computer.org

COMPUTER PROFESSIONALS Central NJ IT Consulting company requires candidates for following positions at their primary Princeton , NJ location, Java Developer: To Design, develop & implement business software apps using Java & J2EE, Candidates must have 1 yr. of mandatory exp in related field. All positions require: MS in CS/Engineering/Info systems/Business or related. BS degree + 5yrs exp can be substituted for the MS reqirmt Any combination of foreign edu + related exp equivalent to a US Masters, or any combination of foreign edu +related exp equivalent to a BS Degree will be accepted. Travel to several unanticipated locations all over US & might involve relocation consistent w/client reqirmts & State & Local reqirmts. Mail your resume to: eVantage Solutions Inc., Inc, Attn: HR, 3 Independence way, STE 209, Princeton NJ 08540.

LECTURER IN COMPUTING AND MATHEMATICAL SCIENCES, SEPTEMBER 2017. The Department of Computing and Mathematical Sciences (CMS) at the California Institute of Technology invites applications for the position of Lecturer in Computing and Mathematical Sciences. This is a (non-tenure-track) career teaching position, with full-time teaching responsibilities. The start date for the position is September 1, 2017 and the initial term of appointment can be up to three years. The lecturer will teach introductory computer science courses including data structures, algorithms and software engineering, and will work closely with the CMS faculty on instructional matters. The ability to teach intermediate-level undergraduate courses in areas such as software engineering, computing systems or compilers is desired. The lecturer may also assist in other aspects of the undergraduate program, including curriculum development, academic advising, and monitoring research projects. The lecturer must have a track record of excellence in teaching computer science to undergraduates. In addition, the lecturer will have opportunities to participate in research projects in the department. An advanced degree in Computer Science or related field is desired but not required. Please view the application instructions and apply on-line at https://applications.caltech.edu/job/cmslect The California Institute of Technology is an Equal Opportunity/Affirmative Action Employer. Women, minorities, veterans, and disabled persons are encouraged to apply.

SOFTWARE ENGINEER: Design & devp server-side sw in Java, Ruby & C++ for next-gen visual analytics & collab suite in MVC design patterns & devp frameworks, incl Spring, Hibernate, & Ruby on Rails. Req BS in Comp Sci, IS, MIS, or rtd, & 6 mos exp in: design & devp server-side sw in Java & C++; devp sw apps in MVC patterns & frameworks, incl Spring & Hibernate; build high-perf backend, web-based analytics apps, & RESTful web svcs; prog w/ reltnl dbase, file sys, concurrency, & multithread; produce design docs, write test cases, & perf unit & integ test for sw; & analyze code for bugs, & perform trblsht & debug. Position at Tableau Software in Seattle, WA. To apply, go to http://bit.ly/2bRvyTj.

TECHNOLOGY

Oracle America, Inc.

has openings for

PRODUCT MANAGER, SOLUTIONS MANAGMENT

positions in **Columbia, MD**.

Job duties include: Plan, initiate, and manage information technology (IT) projects. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to angela.andersen@oracle.com, referencing 385.18285.

Oracle supports workforce diversity.

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Westminster, CO**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to mike.sells@oracle.com, referencing 385.19692.

Oracle supports workforce diversity.

PRODUCT MANAGER

Oracle America, Inc.

has openings for

PRODUCT MANAGER

positions in **Redwood Shores, CA**.

Job duties include: Participate in all software and/or hardware product development life cycle activities. Move software products through the software product development cycle from design and development to implementation, testing and/or marketing. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to Navin.thadani@oracle.com, referencing 385.20854.

Oracle supports workforce diversity.

TORINO, ITALY

July 4-8, 2017

IEEE · IEEE computer society

COMPSAC 2017

Building Digital Autonomy for a Sustainable World

Hosted by the Politecnico di Torino, Turin, Italy

July 4-8

https://www.computer.org/web/compsac2017

COMPSAC is the IEEE Computer Society Signature Conference on Computers, Software, and Applications. It is a major international forum for academia, industry, and government to discuss research results and advancements, emerging problems, and future trends in computer and software technologies and applications. The technical program includes keynote addresses, research papers, industrial case studies, plenary and specialized panels, fast abstracts, a doctoral symposium, poster sessions, and a number of workshops and tutorials on emerging and important topics. The theme of the 41st COMPSAC is *Building Digital Autonomy for a Sustainable World*.

Our world changes constantly, driven by the desire to make our lives more convenient, safe and enjoyable. Once the lore of science-fiction, we now share our planet with billions of automated processes, functioning under the control of computer systems we have developed. As we enjoy the advantages they provide, we can't help but wonder: what will be the next major technical innovations and where might they lead us? We've seen conventional automobiles become self-driving cars, human-directed systems become adapted ones, and human services enhanced by robotic services. Clearly, the movement from physical to cyber-enhanced domains is rapid and increasing, calling upon ever-more exciting innovations in computers, software, and applications.

COMPSAC 2017 will provide a forum for in-depth presentations and discussions of the technical challenges, successes, and failures of moving from traditional, person-centered and directed activities and services to those that are the focus of autonomous systems. Sessions will include topics and issues related to autonomous computing, wearable computing, the internet-of-things, social networking, cross-domain data fusion, privacy, security and surveillance, cloud computing, big data, physiological computing, self-aware and self-expressive systems, adaptive learning and teaching, and emerging architectures and network issues that affect all these developing technology-driven innovations.

Authors are invited to submit original, unpublished research work, as well as industrial practice reports. Simultaneous submission to other publication venues is not permitted. In accordance with IEEE policy, submitted manuscripts will be checked for plagiarism; instances of alleged misconduct will be handled according to the IEEE Publication Services and Product Board Operations Manual. Detailed instructions for electronic paper submission, panel, workshop, and tutorial proposals, fast abstracts, industry papers, poster papers, doctoral symposium, and the review process are available at **https://www/computer.org/web/compsac2017**.

IMPORTANT DATES

Workshop Proposals Due:
15 October 2016
Workshop Proposal Notification:
15 November 2016
Main Conference Papers Due:
13 January 2017
Main Conference Notification:
27 March 2017
Workshop Papers Due:
10 April 2017
Workshop Paper Notification:
25 April 2017
Camera Ready and Registration Due:
9 May 2017

2017 ORGANIZERS

Advisory Committee Chair:
Sorel Reisman, California State University, USA
Contact: sreisman@computer.org
Steering Committee Chair:
Sheikh Iqbal Ahamed, Marquette University, USA
Contact: sheikh.ahamed@marquette.edu
General Chairs:
Claudio Demartini, Politecnico di Torino, Turin, Italy
Tom Conte, Georgia Institute of Technology, USA

Program Chairs in Chief:
Lin Liu, Tsinghua University, China
William Claycomb, Carnegie Mellon University, USA
Motonori Nakamura, NII, Japan
Edmundo Tovar, Universidad Politecnica de Madrid, Spain
Workshop Chairs:
Chung-Horng Lung, Carleton University, Canada
Hiroki Takakura, NII, Japan
Ji-Jiang Yang, Tsinghua University, China
Stelvio Cimato, University of Milan, Italy